Towards circular and asymmetric cooperation in a multi-player Graph-based Iterated Prisoner's Dilemma

Tangui Le Gléau, Xavier Marjou, Tayeb Lemlouma, Benoit Radier

Towards circular and asymmetric cooperation in a multi-player Graph-based Iterated Prisoner's Dilemma

Tangui Le Gléau[1,2], Xavier Marjou[1], Tayeb Lemlouma[2] and Benoit Radier[1]
[1]*Orange Labs, Lannion, France*
[2]*IRISA, Lannion, France*
{*tangui.legleau, xavier.marjou, benoit.radier*}*@orange.com, tayeb.lemlouma@irisa.fr*

Keywords: Game theory, non-cooperative games, iterated prisoner's dilemma, tit-for-tat

Abstract: In collaborations involving multiple actors, it is well known that tensions between individual interest and global welfare can emerge: actors are personally incentivized to have selfish behavior whereas mutual cooperation may provide a better outcome for all. Known as social dilemmas, these cooperation paradigms have aroused renewed interest in solving social issues, particularly in environmental and energy issues. Hybrids methods with Reinforcement Learning (RL) policies and Tit-for-Tat (TFT) strategies have proven successful to identify fruitful collaboration in complex social dilemmas. However, there are also many situations, where cooperation cannot always be given back directly, and has instead to be carried out through one or more intermediary actor(s). This specificity ruins win-win approaches like TFT. To address this specificity, we introduce a Graph-based Iterated Prisoner's Dilemma: a N-player game in which the possible cooperation between players is modeled by a weighted directed graph. In addition to this new paradigm, we propose a graph-based TFT algorithm that we evaluate on multiple scenarios and compare to other algorithms. Our experiments show that leveraging a graph-based structure in the original TFT algorithm allows it to spread favor better collaboration synergies in most situations.

1 INTRODUCTION

Humanity and industry face numerous issues of cooperation where various self-interested actors intervene. Sometimes, while there is no direct personal incentive to cooperate, mutual cooperation can provide better outcomes for all actors. Such situations involving multiple non-cooperative intelligent agents exist or will exist in various areas where resources are scarce like energy among electricity, road traffic among independent autonomous vehicles, and decentralized multi-agent systems (such as machine learning on private devices). Faced with the proliferation of these intelligent multi-agent situations and faced with social and environmental issues such as a possible upcoming shortage of energy and natural resources, it is increasingly vital to focus more on the studies of learning agents in non-cooperative games [Hager et al., 2019].

This category of games where players have no incentive to cooperate despite optimal mutual cooperation is known as social dilemmas. For decades, matrix games have been very popular: with atomic actions (cooperation/defection), three kinds of canonical games have emerged. They all have in common point at least one incentive to choose defection. It can either be fear (the fear of being exploited by a defector while cooperating) like in trust problems known as the Stag Hunt, or greed (the temptation to exploit a cooperator) in the Chicken Game. At last, when both fear and greed incentivize players to defect, they face the famous Prisoner's Dilemma (PD). Social dilemmas have been well studied for many years. The notion of general-sum non-cooperative games is quite ancient [Nash, 1951] as well as the formulation of the PD [Flood, 1958]. Therefore, numerous real life situations have been modeled by such dilemmas: from climate change and environmental security [Soroos, 1994] to institutions negotiation with N-person version of the PD as well as tax policies [Zheng et al., 2020].

If the formulation of this problem is rather simple, the issue of efficient behaviors facing such dilemmas has raised a huge interest for decades. In a seminal work, [Axelrod and Hamilton, 1981] tackles the question of the Iterated Prisoner's Dilemma (IPD) and proposes to oppose various agents in a tournament where they should maximize their rewards playing successively a PD. According to Axelrod, the key of success

rests on four properties: a "good" agent should be nice, provokable, forgiving and clear to understand. The winner of IPD tournament was the algorithm Tit-for-Tat (TFT) introduced by Anatol Rapoport [Rapoport et al., 1965]: this simple algorithm (essentially reproducing the behavior of its partner) fulfills the four properties, manages to encourage cooperation and to be robust to defection. Subsequently, variants of TFT [Verhoeff, 1998] or alternatives [Nowak and Sigmund, 1993] have also fueled interest in social dilemmas. Reinforcement Learning (RL) methods have also been used to study the IPD [Izquierdo et al., 2008].

Later, the emergence of Deep Reinforcement Learning [Mnih et al., 2015] brought a new interest since the training of complex policies has made possible the study of cooperation in more realistic social dilemmas [Leibo et al., 2017, Jaques et al., 2019]. However, [Lerer and Peysakhovich, 2017] has shown empirically that a Tit-for-Tat strategy remains necessary to help RL policies not to fall in Nash equilibrium (mutual defection), which suggests that studying TFT keeps a huge interest, in particular to complement RL policies in more general non-cooperative games.

Tit-for-Tat has been well studied and performs well. However in some realistic situations, direct cooperation may be less optimal or even impossible. Indeed, sometimes cooperation has to go through one or more indirect cooperator(s). Whereas it exists an optimal multi-agent collaboration, the cooperation is now not necessary bilateral and then the basic TFT strategies are inefficient.

To address and study this asymmetry and possibly enable circular cooperation, we introduce the Graph-based Iterated Prisoner's Dilemma (GIPD), a novel kind of N-player Prisoner's Dilemma with a graph-structure to better generalize the complexity of possible cooperation between players. The key element of our formalism is a weighted directed graph which sets the maximal authorized cooperation for each pair of players. To tackle this new paradigm, we extend the classic TFT to create a novel algorithm: the Graph-based Tit-for-Tat (GTFT). To address the asymmetry and weighted cooperation, our extension adopts a flow network approach to detect and maintain cycles of cooperation within the players.

We simulate our GTFT on dilemmas involving various patterns of circularity, and compare it to some classic TFT. We also conduct some experiments on the choice of the TFT functions involved in GTFT. We study the relevance of the graph-based structure and the choice of parameters with five designed social metrics.

In section 2, we define social dilemmas in particular the continuous PD. Before proposing our approach of GTFT in section 5, we discuss about other formalisms in section 3 for the N-player IPD and introduce our model of GIPD in section 4. At last, methods, results and discussions are presented in sections 6 and 7.

2 DEFINITIONS OF SOCIAL DILEMMAS

Social dilemmas are multi-player games where at least one of its Nash equilibria is not optimal. The literature describes them as situations where multiple players can choose between cooperation and defection in a game where mutual cooperation is optimal whereas they all are incentivized to defect. In this section, we recall the formalism of those social dilemmas in particular the so-called Prisoner's Dilemma (PD) and its continuous version.

2.1 Classic Social Dilemma

Matrix games with two players provide a basic framework to introduce social dilemmas [Axelrod and Hamilton, 1981]: Two players A and B choose between cooperation and defection, and they receive one of the four possible payoffs given by Table 1: they are denoted in the literature by R (Reward for mutual cooperation), S (Sucker for being exploited by a defector), P (Punishment for mutual defection) and T (Temptation to exploit a cooperator).

Table 1: Payoffs in a 2-player social dilemma

	Cooperate	Defect
Cooperate	(R,R)	(S,T)
Defect	(T,S)	(P,P)

This matrix game is defined as a social dilemma if the four payoffs verify the following inequalities [Macy and Flache, 2002]:

1. $R > P$: Mutual cooperation is better than mutual defection
2. $R > S$: Mutual cooperation is better than exploitation
3. At least one of these two inequalities:
   - $T > R$: Greed (3a)
   - $P > S$: Fear (3b)
4. $R > \frac{1}{2}(S+T)$: Mutual cooperation is better than equiprobable different choice (4)

The last condition (4) is relevant in the iterated version of the game to avoid that the rotation Cooperate/Defect becomes optimal. In the literature, there are three kinds of social dilemma depending of which incentive of inequality (3) is verified. In this paper, we focus on the case of Prisoner's Dilemma (where greed (3a) and fear (3b) are verified). The often used payoffs $S < P < R < T$ are given by $S = 0, P = 1, R = 3$ and $T = 5$ [Axelrod and Hamilton, 1981]. This particular game admits one Nash Equilibrium which is (Defection, Defection) whereas the optimal outcome is (Cooperation, Cooperation).

2.2 Continuous Prisoner's Dilemma

The classic PD can be extended to a continuous version by extrapolating a payoff with the four values of Table 1. Let us say that two players A and B play a dilemma. Instead of choosing a discrete action between Cooperate and Defect, they now can choose a continuous cooperation degree $a \in [0,1]$ and $b \in [0,1]$ (0 for total defection and 1 for total cooperation). [Verhoeff, 1998] proposes to extrapolate the values of S, P, R and T as by the gain function G follows:

$$\begin{aligned} &G : [0,1] \times [0,1] \to \mathbb{R} \\ &G(x,y) = xyR + (1-x)(1-y)P + x(1-y)S + (1-x)yT \end{aligned} \tag{1}$$

We then define the payoffs of a continuous Prisoner's Dilemma between two players A and B choosing respectively actions a and b by $V_A, V_B = V(a,b)$, with:

$$\begin{aligned} &V : [0,1] \times [0,1] \to \mathbb{R} \times \mathbb{R} \\ &V(a,b) = G(a,b), G(b,a) \end{aligned} \tag{2}$$

For clarity, we can check that with discrete actions (i.e. $a, b \in \{0,1\}^2$), we find the four payoffs in the discrete case given in the Table 1:

- $V(0,0) = P, P$
- $V(1,0) = S, T$
- $V(0,1) = T, S$
- $V(1,1) = R, R$

3 RELATED WORK

Social dilemmas, and in particular the Prisoner's Dilemma, have been studied for decades. They have their roots in the Nash notion of non-cooperative games [Nash, 1951] and then have been properly introduced in the Axelrod's tournament of Iterated Prisoner's Dilemmas [Axelrod and Hamilton, 1981]. The winner of this seminal tournament was the Tit-for-Tat strategy [Rapoport et al., 1965], a rather simple strategy basically consisting in reproducing the opponent choice. Then, numerous improvements have been proposed to the original TFT: [Wu and Axelrod, 1995] tried to cope with the noise of TFT in IPD, [Beaufils et al., 2001] proposed a memory-based TFT. Other novel strategies similar to the TFT have been introduced e.g. win-stay and lose-shift [Nowak and Sigmund, 1993]. Reinforcement Learning techniques have also been adopted to train agents to face the IPD: [Matignon et al., 2012] conducted a study on decentralized multi-agent Q-learning in cooperative and non-cooperative games, [Matignon et al., 2007] proposed the hysteretic Q-learning to cope with non-stationarity of non-cooperative agents. More recently, [Lin et al., 2020] conducted a very complete tournament of bandits, RL and TFT agents.

Moreover, variants of Prisoner's Dilemma have emerged: [Verhoeff, 1998] introduced the continuous PD and some corresponding adapted strategies of TFT ; [Hamburger, 1973] proposed a formalism of N-Iterated Prisoner Dilemmas (NIPD) in which payoffs of players are based on the number of cooperators. Some experiments have been conducted on NIPD [Yao, 1996], and in particular with Reinforcement Learning [Agudo and Fyfe, 2011]. Some studies also tackled the problem of graphs with Prisoner's Dilemmas [Ashlock, 2007], [Luo et al., 2010]. Their models simulate situations of multiple players placed in a (undirected) graph in which they can play a PD only with their neighbors. Our approach differs greatly in a sense that maximal authorized cooperation between players is given by a directed and weighted graph. Thus, the cooperation between each pair of players is directed and weighted (in particular non-existent) and can enable non-symmetrical and even circular situations.

To finish, recently social dilemmas have known a new interest with the emergence of deep RL: [Leibo et al., 2017] introduced sequential social dilemmas in the form of more realistic games in which RL agents are trained to cooperate between each other. [Lerer and Peysakhovich, 2017] showed empirically that RL agents fall in non-optimal Nash Equilibrium, and that TFT was necessary to help RL policies to mutually cooperate and maintain cooperation. The solution mixing TFT strategies and RL policies they brought is very promising, however their solution doesn't cover asymmetric and circular situations.

4 GRAPH-BASED ITERATED PRISONER'S DILEMMA

In this section, we propose to extend the N-player Iterated Prisoner's Dilemma with a graph structure. The main idea is that maximal authorized cooperation within each ordered pair of players is given by a weighted directed graph.

4.1 N-player Prisoner's Dilemma

In this section, we describe the model used for the N-player PD, that we define here without graph extension. We consider a decentralized model, i.e. the N players play one continuous PD between each other: each player i chooses a continuous cooperation degree $c_{ij} \in [0,1]$ towards every other player j ($c_{ij} = 0$ for total defection and $c_{ij} = 1$ for total cooperation). In other words, each player i chooses a vector of cooperation $\overrightarrow{C_i} = (c_{ij})_{j \in [\![1,N]\!]}$ and all decisions of players form the matrix $C = (c_{ij}) \in [0,1]^{N \times N}$. Note that we adopt the convention that one player cannot cooperate with oneself: $\forall i, c_{ii} = 0$. After the decision of cooperation degrees, the $N(N-1)/2$ continuous PD made simultaneously within each pair of players (i,j) give to them the payoffs given by 2.2. Therefore, each player i receives the total payoff:

$$V_i = \sum_{j \neq i} c_{ij}c_{ji}R + (1-c_{ij})(1-c_{ji})P + c_{ij}(1-c_{ji})S + (1-c_{ij})c_{ji}T \tag{3}$$

4.2 Graph-based Iterated Prisoner's Dilemma (GIPD)

We introduce our formalism of GIPD that we define by :

- a number N of players
- a weighted directed graph $\mathcal{G}_{max}$ of maximal cooperation defined by a weighted adjacency matrix $\mathcal{C}_{max} \in [0,1]^{N \times N}$
- a vector $D_{max} \in (\mathbb{R}^+)^N$ of maximal cooperation effort
- a number of steps (or rounds) T_{max}

In this game, at each step t, each agent i can choose $N-1$ cooperation degrees c_{ij}^t with two constraints: first, the maximal value for c_{ij}^t is fixed by the graph $\mathcal{G}_{max}$, and secondly its outgoing flow cannot exceed $D_{max}^i : C_i^{t+} = \sum_{j \neq j} c_{ij}^t \leq D_{max}[i]$. In practice, the first constraint is verified by clipping the total choice matrix $C^t = (c_{ij}^t)$ by the adjacency matrix of maximal cooperation graph $\mathcal{G}_{max} : C^t \leftarrow \min(C^t, \mathcal{C}_{max})$. It means for example that if $\mathcal{C}_{max}[a,b] = 0$, player a cannot cooperate with b, so even if a chooses a degree $c_{ab} = 1$, the effective degree will in fact be equal to 0. To ensure the second constraint, each row $\overrightarrow{C_i^t}$ is normalized by the factor $\min(1, \frac{D_{max}[i]}{C_i^{t+}})$. When the effective matrix $C^t = (c_{ij}^t)$ is computed, the agents receive the payoffs determined by the formulation 3.
In this game, we assume that all players can observe the full effective choice matrix of the previous step C^{t-1}, that they have the knowledge of $\mathcal{C}_{max}$ and D_{max}, but T_{max} is unknown to them. This formalism of social dilemma allows to model the classic prisoner's dilemma (Figure 1a) as well as what we call circular dilemmas i.e. the presence of a cyclic flow in the weighted directed graph of maximal cooperation $\mathcal{G}_{max}$ (e.g. Figures 2c or 2d). For better intuition, we provide in Figure 2 some real examples related to the cooperation graphs represented in Figure 1.

4.3 Examples

Although applications of this formalism are out of scope of the paper, we nevertheless provide below (Figure 2) some didactic examples of dilemmas related to the GIPD presented in Figure 1. The edge weights of the maximal cooperation graphs can be roughly viewed in this case as proportional to the payoffs sum of the helper and the "receiver".

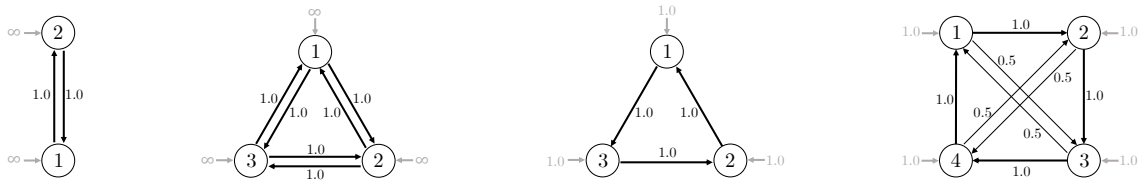
The above example of dilemmas represents the numerous situations where multiple actors can share some items (or services) to homogenize their resources (or energy) by mutually giving their surplus to others.

5 GRAPH-BASED TIT-FOR-TAT

In this section, we update the classic Tit-for-Tat algorithm to adapt it to the GIPD (defined in 4.3). We start by recalling the principle of the classic Tit-for-Tat algorithm, in particular in a continuous version, and then we detail the novel Graph-based Tit-for-Tat algorithm we propose.

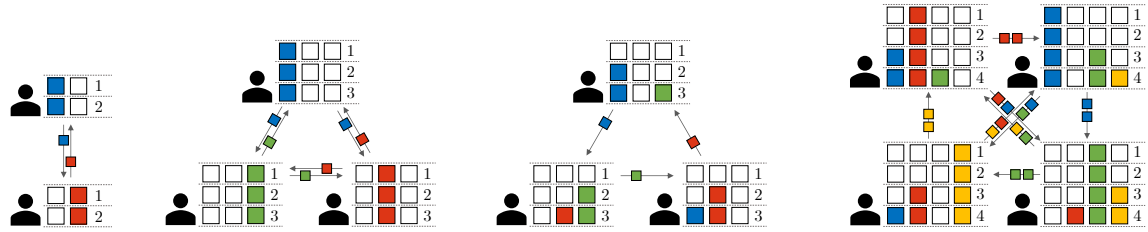
5.1 Definition of Tit-for-Tat

We assume that a TFT function f_{TFT}^k with k steps of memory is a function indicating at each time step t what a player A (facing a player B) should choose as cooperation degree $a_t \in [0,1]$ according to the k previous cooperation degrees:



(a) (b) (c) (d)

Figure 1: Some examples of Graph-based Prisoner's Dilemmas: (1a) is the classic 2-player PD. (1b) is a 3-player full version of PD. (1c) is a 3-player circular PD and (1d) is a 4-player circular PD with an alternative with four 3-player cycles of cooperation (less optimal : flow of 0.5 instead of 1.0). The values of maximal cooperation effort of D_{max} are indicated in gray.



(a) (b) (c) (d)

Figure 2: A game where N players can decide to cooperate by sharing their items or defect by keeping them. The items have a decreasing marginal utility (e.g. $N-k+1$ for the k^{th} same item) thereby conferring the game a dilemma property: the optimal outcome is the mutual cooperation which is exchanging equitably items, but greed and fear lead to the non-optimal mutual defection. According to the topology of resources of agents, circular situations can emerge (2c and 2d).

$$\begin{aligned} &f^k_{TFT} : \mathbb{N} \times [0,1]^{2k} \to [0,1] \\ &f^k_{TFT}(t, a_{t-1}, b_{t-1}, ..., a_{t-k}, b_{t-k}) = a_t \end{aligned} \tag{4}$$

The majority of TFT algorithms needs only the previous step ($k = 1$) but some approaches needs more steps, like the gradual behavior with memory proposed by [Beaufils et al., 2001] and we can theoretically envisage the case $k = +\infty$ with the use of models learning the history (Recurrent Neural Networks).

For example, the original Tit-for-Tat with discrete choice (cooperation/defection i.e. a cooperation degree in $\{0,1\}$) is defined by cooperating on the first move (cooperation equal to 1) then doing whatever the other player did on the previous move. For better understanding of our notations, we detail the formalism of the classic TFT:

$$\begin{aligned} &\text{TFT} : \mathbb{N} \times \{0,1\} \to \{0,1\} \\ &\text{TFT}(t, b_{t-1}) = \begin{cases} 1, & \text{if } t = 0, \\ b_{t-1}, & \text{if } t > 0. \end{cases} \end{aligned} \tag{5}$$

5.2 Continuous Tit-for-Tat

To address the continuous version of an IPD (defined in 2.2), some continuous versions of Tit-for-Tat exist. We gather propositions of [Verhoeff, 1998] and [Le Gléau et al., 2020] as well as a contribution (the stochastic incentive) in the following expression:

$$\begin{aligned} &\text{TFT}_{\alpha,\beta,\gamma,r_0,c_0} : \mathbb{N} \times [0,1]^2 \to [0,1] \\ &\text{TFT}_{\alpha,\beta,\gamma,r_0,c_0}(t, a_{t-1}, b_{t-1}), r_t = \\ &\begin{cases} c_0, r_0 & \text{if } t = 0 \\ \alpha a_{t-1} + (1-\alpha)(r_{t-1} + (1-r_{t-1})b_{t-1}), & \\ [r_{t-1} + \beta(b_{t-1} - a_{t-1})]^+ + r_0 X_\gamma & \text{if } t > 0 \end{cases} \end{aligned} \tag{6}$$

In this expression, α is an inertia coefficient to smooth reaction. r_t is a coefficient to incentivize cooperation (r_0 is the initial value of r_t). β is the adaptive coefficient to make r_t dynamic: it increases (resp. decreases) when partner cooperation increases (resp. decreases). The objective of that adaptive coefficient is to gain safety. At last, X_γ is a Bernoulli variable of probability γ, it means that $r_0 X_\gamma$ is equal to r_0 with probability γ and 0 otherwise. This stochastic parameter allows to stimulate incentive in case of null value of r_t for all players. At last, for our simulations, we use an null initial degree $c_0 = 0.0$. To sum up the notations, we consider three kinds of TFT functions:

- TFT_alpha: we omit adaptive and stochastic parameters ($\beta = \gamma = 0$), i.e. the incentive coefficient is constant $r_t = r_0$
- TFT_beta: we omit stochastic parameter ($\gamma = 0$)
- TFT_gamma: we use all parameters

5.3 Graph-based Tit-for-Tat

We introduce the Graph-based Tit-for-Tat[1]. The goal of this algorithm is to extend the classic TFT in particular to cope with asymmetry and the existence of cycles of cooperation. Our idea relies on viewing the cooperation graph as a flow network [Ford Jr, 1956] and finding the maximum flow which makes a cycle. We detail our algorithm denoted GRAPHTFT for the point of view of player k. The principle is that GRAPHTFT(k) has inner variables that it updates at each step: a cooperation graph $\mathcal{C}_k$, a graph which allows to penalize defectors and a source flow $\mathcal{D}_k$, an amount of cooperation that it is ready to provide. In the following, we consider that the algorithm uses a TFT function $f_{TFT} : [0,1] \times [0,1] \to [0,1]$ such that $a^{t+1} = f_{TFT}(a^t, b^t)$. This function allows to update over time a coefficient a according to an observed coefficient b fulfilling the objectives of TFT spirit. GRAPHTFT(k) is divided into several phases, at each step:

1. For each other player j, update $\mathcal{C}_k[k,j]$ with f_{TFT} according to the difference between what j "received" and what he "gave" at previous step.
2. Update the source flow $\mathcal{D}_k$ with f_{TFT} according to the difference between what k (oneself) "has received" and what k (oneself) "had given" at previous step.
3. Create a flow network $\mathcal{F}$, whose capacities are given by $\mathcal{C}_k$ with a source vertex directed towards the vertex of k (oneself) with capacity $\mathcal{D}_k$ and all edges of $\mathcal{C}_k$ initially directed towards vertex k artificially redirected towards a sink vertex. Thus, we have a flow network allowing to find the maximum cyclic flow (i.e. from k to k). For better understanding, see Figure 3.
4. Compute the maximum flow $\mathcal{R}$ on $\mathcal{F}$, i.e. a sub-graph in $\mathcal{F}$ and extract the next choice of cooperation $\overrightarrow{C_k} \leftarrow \mathcal{R}[k,:]$

5.3.1 Max flow problem

To find the maximum cyclic flow of cooperation, our algorithm converts its inner cooperation graph into a flow network (Figure 3), and finds the maximal flow. We use two kinds of algorithms with polynomial complexity ($O(\Delta N^2)$ where Δ is the discretization number):

- Ford-Fulkerson algorithm [Ford and Fulkerson, 1956]: find the maximal flow (but with shortest path)

[1]For reproducibility, the code has been made available: https://github.com/tlgleo/graph_based_TFT

Algorithm 1: GRAPHTFT (for agent k)

Input: Max cooperation graph $\mathcal{C}_{max}$ and max source flow D_{max} given by the game and a TFT function f_{TFT}
Initialize: $\mathcal{C}_k \leftarrow \mathcal{C}_{max}$, $\mathcal{D}_k \leftarrow D_{max}[k]$
First step: Choose $\forall j \neq k$, $\vec{C}_k[j] \leftarrow f_{TFT}(t=0)$
for $t \in [1, T_{max}]$ **do**
  for *each other agent* j **do**
    From C^{t-1}, compute outgoing flow of cooperation of j : $(C_j^{t-1})^+$
    Execute a TFT on j : $c_{kj}^t = f_{TFT}(c_{kj}^{t-1}, (C_j^{t-1})^+)$
    Modify the inner cooperation graph: $\mathcal{C}_k[k,j] \leftarrow c_{kj}^t \mathcal{C}_{max}$
  end
  From C^{t-1}, compute the incoming flow of cooperation for k : $(C_k^{t-1})^-$
  Update by TFT the next source flow: $\mathcal{D}_k \leftarrow f_{TFT}(\mathcal{D}_k, (C_k^{t-1})^-)$
  Generate a new flow network $\mathcal{F}$ from k to k with a source of capacity $\mathcal{D}_k$ and capacities given by $\mathcal{C}_k$ (see Figure 3)
  From $\mathcal{F}$, extract the sub-graph $\mathcal{R}$ of maximum flow of cooperation
  Choose cooperation degrees from max flow: $\overrightarrow{C_k^t} \leftarrow \mathcal{R}[k,:]$
end

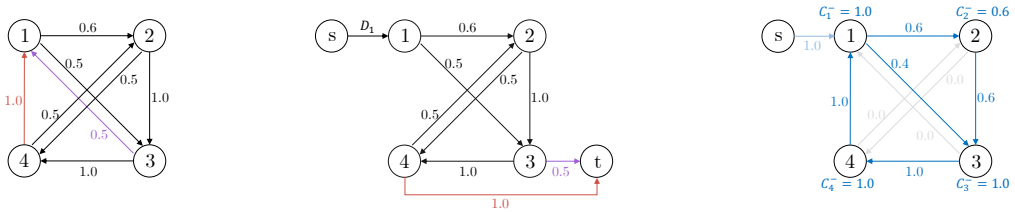
- Min-cost max flow [Orlin, 1997]: a variant of Ford-Fulkerson, with a minimization of a certain cost per chosen edge (for the cost, we choose the inverse of cooperation so that the flow search prefers longest cycles.)

6 METHODS

In this section, we define and detail some methodological points such as our designed metrics as well as the agents and tournaments we use in our simulations.

6.1 Metrics

Let N agents denoted $\vec{\pi} = (\pi_i)$ acting for the N players of a tournament $\mathcal{T}$ during T_{max} steps. We denote $V_i(\mathcal{T}, \vec{\pi})(t)$ the T_{max} payoffs received by the player i and $\overline{V_i(\mathcal{T}, \vec{\pi})}$ the sum of its T_{max} payoffs. Moreover, we define the social welfare $SW(\mathcal{T}, \vec{\pi}, t)$ as the sum of payoffs of the N agents at time t. In the following, we succinctly define the five metrics we design (adapted from [Lerer and Peysakhovich, 2017]).



(a) Step 1: Tit-for-Tat

(b) Step 2: Transformation of the graph 3a into a flow network

(c) Step 3: Finding the max flow sub-graph

Figure 3: Steps to find a cyclic sub-graph of max flow from player 1 in the graph (3a). The graph is transformed into a flow network regarding player 1 with a source s and a sink t in step (3b) and the max flow is extracted in (3c).

First, the Utilitarian metric U measures how close the social welfare is from the optimum which is mutual cooperation (C) compared to the mutual defection (D), and the speed Sp measures how fast the Utilitarian metric reaches its maximal value. Then the incentive-compatibility IC is destined to measure the capacity to incentivize cooperation within other players. We define the incentive-compatibility of the agent π by the difference between what one agent ($i = 0$) receives by cooperating with all other agents π compared to what it would have received by defecting. The safety Sf measures the risk an agent takes by preferring π rather than defection face to defectors ; let us note that since defection is dominant (by definition of dilemma), this metric is always non positive: the higher it is, the more the agent is safe. At last, the forgiveness Fg measures how the social welfare is impacted when a "repentant defector" R begins to cooperate only after τ_0 steps. For the sake of clarity, we denote $x\vec{1} = (x)_i$ the constant joint policy of N or $N-1$ agents choosing x.

$$U(\mathcal{T},\pi,t) = \frac{SW(\mathcal{T},\pi\vec{1},t) - SW(\mathcal{T},D\vec{1},0)}{SW(\mathcal{T},C\vec{1},0) - SW(\mathcal{T},D\vec{1},0)}, \quad Sp(\mathcal{T},\pi) = \frac{1}{\tau U_{max}} \int_0^{\tau} U(\mathcal{T},\pi,t)dt$$

$$IC(\mathcal{T},\pi) = \frac{\overline{V_0(\mathcal{T},(C,\pi\vec{1}))} - \overline{V_0(\mathcal{T},(D,\pi\vec{1}))}}{\overline{V_0(\mathcal{T},C\vec{1})} - \overline{V_0(\mathcal{T},D\vec{1})}}, \quad Sf(\mathcal{T},\pi) = \frac{\overline{V_0(\mathcal{T},(\pi,D\vec{1}))} - \overline{V_0(\mathcal{T},D\vec{1})}}{\overline{V_0(\mathcal{T},D\vec{1})} - \overline{V_0(\mathcal{T},(C,D\vec{1}))}}$$

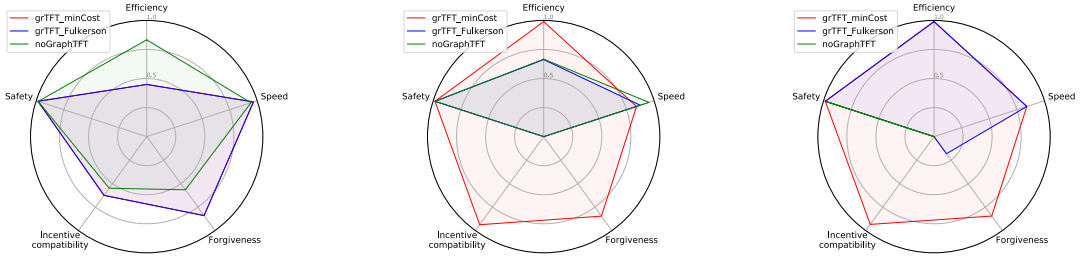
$$Fg(\mathcal{T},\pi) = \frac{1}{T_{max} - \tau_0} \sum_{t=\tau_0}^{T_{max}-1} \left[\frac{SW(\mathcal{T},(R,\pi,...,\pi),t) - SW(\mathcal{T},D\vec{1},\tau_0)}{SW(\mathcal{T},C\vec{1},\tau_0) - SW(\mathcal{T},D\vec{1},\tau_0)} \right]$$

6.2 Agents and tournaments

In our simulations, we compare several agents of TFT. We use as baseline the vanilla continuous TFT and we compare the different graph algorithms of our novel GRAPHTFT. Moreover, we also compare the choice of the TFT function in our algorithm and evaluate the addition of our improvements (Alpha, Beta, Gamma). For our simulations, we created two kinds of tournaments of $N > 2$ players with some patterns of circularity. One is purely circular that we denote CIRC(N): the weight of edge (i, j) is equal to 1.0 if $j = (i+1) \mod N$ and 0.0 otherwise. The other one is double circular DOUBLE(N), the same with an alternative cooperation edge (to study the case when one defector breaks the cooperation cycle): the weight of edge (i, j) is equal to 1.0 if $j = (i+1) \mod N$ or $j = (i+2) \mod N$ and 0.0 otherwise.

7 RESULTS AND DISCUSSION

In this section, we present the results of some simulations where we compare our extension of TFT suited for the GIPD compared to the classic TFT. We first study the relevance of the graph structure of our GTFT and then the impact of the choice of the TFT function.

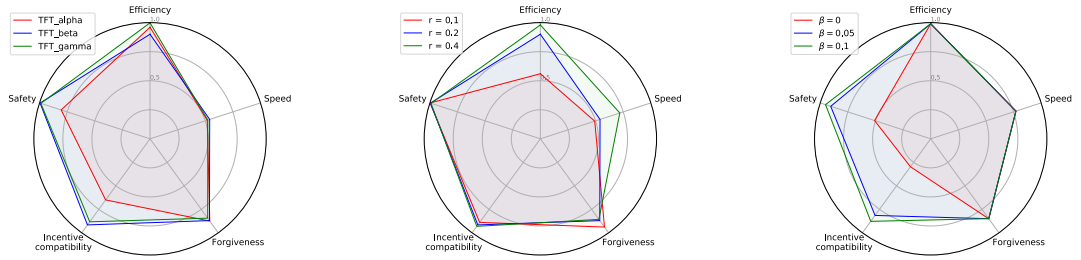


(a) Double Circular 3 players[1]

(b) Double Circular 4 players[2]

(c) Double Circular 6 players[3]

Figure 4: Impact of the choice of graph processing in the double circular tournament with 3, 4 and 6 players. The Tit-for-Tat algorithm used for all agents of that simulation is the algorithm TFT_beta with parameters ($\alpha = 0.6$, $\beta = 0.6$, $r_0 = 0.7$)



(a) Impact of type

(b) Impact of incentive r_0

(c) Impact of adaptive β

Figure 5: Impact of the choice of TFT function on DOUBLECIRC(6) with GRAPHTFT (with min cost approach). We compare the three kinds of TFT (Figure 5a), then the impact of the initial incentive r_0 (Figure 5b) and the adaptive coefficient β (Figure 5c) on the TFT_beta algorithm.

7.1 Impact of the graph-processing algorithm

First, we focus on the necessity of adding a graph-processing algorithm and study the impact of the chosen algorithm. The three cases we study are:

- noGraphTFT: classic continuous TFT with no graph processing.
- grTFT_Fulkerson: our GRAPHTFT (algo 1) with a Ford-Fulkerson algorithm to find the maximum flow.
- grTFT_minCost: our GRAPHTFT (algo 1) with a min-cost max flow approach: it searches the max flow but with constraint of maximal circulated flow (cooperation through maximum number of players).

We simulate these three kinds of agents in the games DOUBLECIRC with 3, 4 or 6 players and compare the results of the metrics in Figure 4.

[1]A simulation of the cycle issue with 3 players: https://youtu.be/VHhEZ8Wu_XQ or with 5 players: https://youtu.be/J-weuvOkkBc

[2]A simulation with 4 players: https://youtu.be/s08j24LMr0U

[3]A simulation with 6 players: https://youtu.be/AL29LFbh3n8

The observations confirm the intuition that the lack of graph processing in circular situations totally defeats the classic TFT in the 6-player game since there is no possibility of symmetrical cooperation. In the 4-player game, the classic TFT can find a less optimal solution since it exists two less optimal 2-player cycles. At last, the classic TFT can address the 3 players DOUBLECIRC because a symmetrical alternative is possible but a bit less optimal. Regarding the choice of graph processing, the best one is the approach with min cost because it allows to select the most pro-social cycles of cooperation (unlike the Ford-Fulkerson algorithm). It is therefore more suited to incentivize cooperation and is safer to defection. However, when it exists several equivalent optimal cycles (e.g. with 3 or 5 players), both GTFT agents naturally hardly synchronize themselves.

7.2 Impact of choice of Tit-for-Tat function

We study here the choice of Tit-for-Tat function f_{TFT} used in our GRAPHTFT (algo 1). We first evaluate the impact of the choice of TFT among the described algorithms in section 5.2 (TFT_alpha with inertia, TFT_beta with adaptive incentive and TFT_gamma with stochastic incentive). We then study the impact of

initial incentive coefficient r_0 and adaptive coefficient β with the TFT_beta.

In Figure 5a, we can observe that the adaptive coefficient β is interesting since it makes TFT_beta and TFT_gamma safer and more incentive compatible. TFT_gamma seems to be a bit more efficient in a sense that the outcome is more optimal. Regarding the impact of parameters of the TFT_beta, we can remark that a higher initial incentive coefficient r_0 allows to reach a more optimal outcome more rapidly without reducing safety and incentive-compatibility. Regarding the impact of adaptive coefficient β, we can observe that a tiny non-null coefficient is sufficient to make the agent safer and more incentive compatible.

7.3 Results Summary

Although our GTFT is not perfectly optimal in ambiguous situations (with multiple optimal cycles), we can conclude that the key point of our experiments is the importance of a graph-processing algorithm. The min-cost max flow is the best approach since it is more incentive-compatible. Regarding the choice of TFT function, the TFT_beta function is clearly safer than TFT_alpha while TFT_gamma is slightly more efficient.

8 CONCLUSION

In this paper, we introduced a novel paradigm for the N-player Prisoner's Dilemma where maximal cooperation between agents is induced by a weighted directed graph. This new model is particularly suited to address the asymmetry of cooperation and in particular the circular social dilemmas: a specific situation of dilemma where players can form a cycle of cooperation in which no player can cooperate with its "helper". We showed that classic solutions like Tit-for-Tat strategies cannot solve properly this specific issue, we therefore also proposed in the paper a Graph-based Tit-for-Tat which generalizes the classic TFT with a flow network approach. We evaluated this new algorithm in some scenarios and compare it to some baselines. As major conclusions, we can observe that adding a graph processing in the TFT is relevant since our GTFT outperforms the original TFT in most of situations. As further works, it could be very interesting to address the ambiguous cases with multiple equivalent optimal cycles.
We recall our main contributions:

- We introduced and formalized a novel Graph-based Iterated Prisoner's Dilemma: a formalism able to generalize the N-player IPD involving asymmetrical or circular cooperation.
- We designed and formalized several social metrics adapted to this GIPD.
- We constructed a novel Graph-based Tit-for-Tat able to cope with circular cooperation, it is based on continuous TFT and max-flow algorithms.

We are convinced that this new GTFT paradigm which solves circular dilemmas should offer a lot of perspectives particularly in addition to the recent techniques mixing RL and TFT. Finally, in view of the expectations regarding the digital sobriety and the ethical stakes of artificial intelligence, we reiterated the importance of focusing urgently on non-cooperative games, and striving to include this kind of paradigm in the design of our future intelligent systems.

REFERENCES

Agudo, J. E. and Fyfe, C. (2011). Reinforcement learning for the n-persons iterated prisoners' dilemma. In *2011 Seventh International Conference on Computational Intelligence and Security*, pages 472–476. IEEE.

Ashlock, D. A. (2007). Cooperation in prisoner's dilemma on graphs. In *2007 IEEE Symposium on Computational Intelligence and Games*, pages 48–55. IEEE.

Axelrod, R. and Hamilton, W. D. (1981). The evolution of cooperation. *science*, 211(4489):1390–1396.

Beaufils, B., Delahaye, J.-P., Mathieu, P., et al. (2001). Adaptive behaviour in the classical iterated prisoner's dilemma. In *Proc. Artificial Intelligence & Simul. Behaviour Symp. on Adaptive Agents & Multi-Agent Systems*. Citeseer.

Flood, M. M. (1958). Some experimental games. *Management Science*, 5(1):5–26.

Ford, L. R. and Fulkerson, D. R. (1956). Maximal flow through a network. *Canadian journal of Mathematics*, 8:399–404.

Ford Jr, L. R. (1956). Network flow theory. Technical report, Rand Corp Santa Monica Ca.

Hager, G. D., Drobnis, A., Fang, F., Ghani, R., Greenwald, A., Lyons, T., Parkes, D. C., Schultz, J., Saria, S., Smith, S. F., et al. (2019). Artificial intelligence for social good. *arXiv preprint arXiv:1901.05406*.

Hamburger, H. (1973). N-person prisoner's dilemma. *Journal of Mathematical Sociology*, 3(1):27–48.

Izquierdo, S. S., Izquierdo, L. R., and Gotts, N. M. (2008). Reinforcement learning dynamics in social dilemmas. *Journal of Artificial Societies and Social Simulation*, 11(2):1.

Jaques, N., Lazaridou, A., Hughes, E., Gulcehre, C., Ortega, P., Strouse, D., Leibo, J. Z., and De Freitas, N. (2019). Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International Conference on Machine Learning*, pages 3040–3049. PMLR.

Le Gléau, T., Marjou, X., Lemlouma, T., and Radier, B. (2020). Game theory approach in multi-agent resources sharing. In *25th IEEE Symposium on Computers and Communications (ISCC)*.

Leibo, J. Z., Zambaldi, V., Lanctot, M., et al. (2017). Multi-agent reinforcement learning in sequential social dilemmas. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 464–473.

Lerer, A. and Peysakhovich, A. (2017). Maintaining cooperation in complex social dilemmas using deep reinforcement learning. *arXiv preprint arXiv:1707.01068*.

Lin, B., Bouneffouf, D., and Cecchi, G. (2020). Online learning in iterated prisoner's dilemma to mimic human behavior. *arXiv preprint arXiv:2006.06580*.

Luo, L., Chakraborty, N., and Sycara, K. (2010). Prisoner's dilemma in graphs with heterogeneous agents. In *2010 IEEE Second International Conference on Social Computing*, pages 145–152. IEEE.

Macy, M. W. and Flache, A. (2002). Learning dynamics in social dilemmas. *Proceedings of the National Academy of Sciences*, 99(suppl 3):7229–7236.

Matignon, L., Laurent, G. J., and Le Fort-Piat, N. (2007). Hysteretic q-learning: an algorithm for decentralized reinforcement learning in cooperative multi-agent teams. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 64–69. IEEE.

Matignon, L., Laurent, G. J., and Le Fort-Piat, N. (2012). Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems. *Knowledge Engineering Review*, 27(1):1–31.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

Nash, J. (1951). Non-cooperative games. *Annals of mathematics*, pages 286–295.

Nowak, M. and Sigmund, K. (1993). A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature*, 364(6432):56–58.

Orlin, J. B. (1997). A polynomial time primal network simplex algorithm for minimum cost flows. *Mathematical Programming*, 78(2):109–129.

Rapoport, A., Chammah, A. M., and Orwant, C. J. (1965). *Prisoner's dilemma: A study in conflict and cooperation*, volume 165. University of Michigan press.

Soroos, M. S. (1994). Global change, environmental security, and the prisoner's dilemma. *Journal of Peace Research*, 31(3):317–332.

Verhoeff, T. (1998). The trader's dilemma: A continuous version of the prisoner's dilemma. *Computing Science Notes*, 93(02).

Wu, J. and Axelrod, R. (1995). How to cope with noise in the iterated prisoner's dilemma. *Journal of Conflict resolution*, 39(1):183–189.

Yao, X. (1996). Evolutionary stability in the n-person iterated prisoner's dilemma. *BioSystems*, 37(3):189–197.

Zheng, S., Trott, A., Srinivasa, S., Naik, N., Gruesbeck, M., Parkes, D. C., and Socher, R. (2020). The ai economist: Improving equality and productivity with ai-driven tax policies. *arXiv preprint arXiv:2004.13332*.

A Theoretical approach

Our conclusions were mainly experimental. In this section, we propose to study some properties of our policy TFT in a cycle of players (CIRC(K)).

Proposition 1. *(Efficiency) Let a cycle of K players (*CIRC(K)*) who all follow the policy* $TFT_{\alpha,\beta,0,r_0,c_0}$. *Then,* $\forall i \in [0,K-1]$, *with* $j = i+1 \ mod \ K$, *we have* $\forall t, \quad c^t_{ij} = c^t_{\rightarrow}$ *where* $c^t_{\rightarrow}$ *is defined by :*

$$\begin{aligned} &c^t_{\rightarrow} = 1 - Q^t(1-c_0) \\ &avec \ Q = \alpha + (1-\alpha)(1-r_0) \end{aligned} \tag{7}$$

Proof. For the K players, the policy of the incoming flow of cooperation is the same as its outgoing flow, so we have the same function of cooperation degree for all players : $\forall i \in [0,K-1]$, with $j = i+1 \bmod K$, we have $\forall t, \quad c^t_{ij} = c^t_{\rightarrow}$. First, let us focus on r_t. Since $\beta(c^{t-1}_{(i-1)i} - c^{t-1}_{i(i+1)}) = \beta(c^{t-1}_{\rightarrow} - c^{t-1}_{\rightarrow}) = 0$ and $\gamma = 0.0$, we have $\forall t, \ r_t = r_0$. We can rewrite the expression of the sequence $c^t_{\rightarrow}$:

$$\begin{aligned} c^t_{\rightarrow} &= \mathrm{TFT}_{\alpha,\beta,0,r_0,c_0}(t, c^{t-1}_{\rightarrow}, c^{t-1}_{\rightarrow}) \\ &= \alpha c^{t-1}_{\rightarrow} + (1-\alpha)(r_0 + (1-r_0)c^{t-1}_{\rightarrow}) \\ &= [\alpha + (1-\alpha)(1-r_0)]c^{t-1}_{\rightarrow} + (1-\alpha)r_0 \\ &= Qa_{t-1} + R \\ \text{with } Q &= \alpha + (1-\alpha)(1-r_0) \text{ and } R = (1-\alpha)r_0 \end{aligned} \tag{8}$$

It is an arithmetico–geometric sequence of common ratio Q and common difference R. If we note $W = \frac{R}{1-Q}$, the solution of $c^t_{\rightarrow}$ is given by:

$$c^t_{\rightarrow} = Q^t(c_0 - W) + W \tag{9}$$

Let us compute the value of W:

$$\begin{aligned} W &= \frac{R}{1-Q} \\ &= \frac{(1-\alpha)r_0}{1-(\alpha+(1-\alpha)(1-r_0))} \\ &= \frac{(1-\alpha)r_0}{(1-\alpha)(1-(1-r_0))} \\ &= 1 \end{aligned} \tag{10}$$

Therefore, the final expression of $c^t_{\rightarrow}$ is given by:

$$\begin{aligned} &c^t_{\rightarrow} = 1 - Q^t(1-c_0) \\ &\text{with } Q = \alpha + (1-\alpha)(1-r_0) \end{aligned} \tag{11}$$

□

Proposition 2. *(Safety) Let a player A, using the TFT function $TFT_{\alpha,\beta,0,r_0,c_0}$ to choose a_t, he is facing a pure defector B ($\forall t,\ b_t = 0.0$).*
If $\beta \geq \frac{r_0(1-\alpha)}{a_0}$, then it exists τ such that $\forall t \geq \tau,\ r_t = 0.0$ and therefore $\lim_{t\to+\infty} a_t = 0$

Proof. First, we have :

$$\begin{aligned} a_t &= \alpha a_{t-1} + (1-\alpha)(r_t + (1-r_t)b_{t-1}) \\ &= \alpha a_{t-1} + (1-\alpha)r_t \\ &\geq \alpha a_{t-1} \end{aligned} \tag{12}$$

With this geometric sequence, we can find a lower bound for a_t:

$$\forall t,\ a_t \geq a_0\alpha^t \tag{13}$$

Let us find an upper bound for r_t:

$$\begin{aligned} r_t &= r_{t-1} + \beta(b_{t-1} - a_{t-1}) \\ &= r_{t-1} - \beta a_{t-1} \\ &\leq r_{t-1} - \beta a_0 \alpha^{t-1} \end{aligned} \tag{14}$$

Hence,

$$\forall u \geq 1, \quad r_u - r_{u-1} \leq -\beta a_0 \alpha^{u-1} \tag{15}$$

We can add up the terms:

$$\begin{aligned} &\forall t \geq 1, \\ &\sum_{u=1}^{t} r_u - r_{u-1} \leq -\beta a_0 \sum_{u=1}^{t} \alpha^{u-1} \\ &\iff r_t - r_0 \leq -\beta a_0 \frac{1-\alpha^t}{1-\alpha} \\ &\iff r_t \leq r_0 - \beta a_0 \frac{1-\alpha^t}{1-\alpha} \end{aligned} \tag{16}$$

Then, the condition $\beta \geq \frac{r_0(1-\alpha)}{a_0}$ is a sufficient condition such that it exists τ for which $\forall t \geq \tau,\ r_t = 0$.

Finally, $\forall t \geq \tau, \quad a_t = a_\tau \alpha^{(t-\tau)}$ and therefore $\lim_{t\to+\infty} a_t = 0$

□