Introducing decision entrustment mechanism into repeated bilateral agent interactions to achieve social optimality

Citation: Hao, Jianye, and Ho-fung Leung. "Introducing Decision Entrustment Mechanism into Repeated Bilateral Agent Interactions to Achieve Social Optimality." Autonomous Agents and Multi-Agent Systems 29, no. 4 (May 17, 2014): 658–682.

As Published: http://dx.doi.org/10.1007/s10458-014-9265-1

Publisher: Springer US

Persistent URL: http://hdl.handle.net/1721.1/106988

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Noname manuscript No.
(will be inserted by the editor)

Introducing Decision Entrustment Mechanism into Repeated Bilateral Agent Interactions to Achieve Social Optimality

Jianye Hao · Ho-fung Leung

the date of receipt and acceptance should be inserted later

Abstract During multiagent interactions, robust strategies are needed to help the agents to coordinate their actions on efficient outcomes. A large body of previous work focuses on designing strategies towards the goal of Nash equilibrium under self-play, which can be extremely inefficient in many situations such as prisoner's dilemma game. To this end, we propose an alternative solution concept, socially optimal outcome sustained by Nash equilibrium (SOSNE), which refers to those outcomes that maximize the sum of all agents' payoffs among all the possible outcomes that can correspond to a Nash equilibrium payoff profile in the infinitely repeated games. Adopting the solution concept of SOSNE guarantees that the system-level performance can be maximized provided that no agent will sacrifice its individual profits. On the other hand, apart from performing well under self-play, a good strategy should also be able to well respond against those opponents adopting different strategies as much as possible. To this end, we consider a particular class of rational opponents and we target at influencing those opponents to coordinate on SOSNE outcomes. We propose a novel learning strategy TaFSO which combines the characteristics of both *teacher* and *follower* strategies to effectively influence the opponent's behavior towards SOSNE outcomes by exploiting their limitations. Extensive simulations show that our strategy TaFSO achieves better performance in terms of average payoffs obtained than previous work under both self-play and against the same class of rational opponents.

Jianye Hao
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Present Address: Massachusetts Institute of Technology
E-mail: jianye@mit.edu

Ho-fung Leung
Department of Computer Science and Engineering
The Chinese University of Hong Kong
E-mail: lhf@cse.cuhk.edu.hk

Keywords Multiagent Learning · Repeated Games · Socially Optimal Outcomes Sustained by Nash Equilibrium

1 Introduction

Multiagent learning has received extensive attention in the literature and lots of learning strategies [14, 25, 4, 9, 11] have been proposed to coordinate the interactions among agents. The multi-agent learning criteria proposed in [4] require that an agent should be able to converge to a stationary policy against some class of opponent (convergence) and the best-response policy against any stationary opponent (rationality). If both agents adopt rational learning strategies in the context of repeated games and also their strategies converge, then they will converge to the Nash equilibrium of the stage game. Indeed, convergence to Nash equilibrium has been the most commonly adopted goal to pursue within different multiagent environments in the multiagent learning literature, and representative examples include distributed Q-learning in cooperative games [13], minimax Q-learning in zero-sum games [14], Nash Q-learning in general-sum games [10], to name just a few.

Convergence is a desirable property in multiagent systems, however, converging to Nash equilibrium may not be the most preferred since it does not guarantee that the agents can receive their best payoffs. One well-known example is the prisoner's dilemma (PD) game shown in Fig. 1a. By converging to the Nash equilibrium (D, D), both agents obtain the payoff of 1, while they could have received a much higher payoff by coordinating on the non-equilibrium outcome (C, C). To this end, the concept of Pareto-optimal outcomes sustained by Nash equilibrium (POSNE) [3] has been proposed as an alternative solution that the agents should learn to converge to. POSNE outcomes refer to those outcomes that are Pareto-optimal and also correspond to a Nash equilibrium payoff profile when the game is infinitely repeated. For example, the outcome (C, C) in the PD game is a POSNE outcome. Converging to POSNE outcomes is attractive in that it not only solves the inefficiency problem (e.g., the PD game), but also is stable since any deviation from POSNE outcomes can be punished which is guaranteed by the Folk theorem [19]. However, there may exist multiple POSNE outcomes, and it is not clear which POSNE outcome that the agents should learn to converge to. Therefore, in this work, we introduce a more refined solution concept - socially optimal outcome sustained by Nash equilibrium (SOSNE), which represents those outcomes maximizing the sum of all agents' payoffs involved among all the POSNE outcomes. For example, in the prisoner's dilemma game, there is only one SOSNE outcome, i.e., outcome (C, C). Compared with POSNE outcomes, SOSNE outcomes still sustain the desirable property of stability as POSNE outcomes, and can be considered as the optimal POSNE outcomes in terms of maximizing the system level efficiency.

The most commonly adopted interacting framework in multiagent learning literature is two-player repeated games, in which each agent chooses its action

independently and simultaneously each round. To reach POSNE outcomes, Sen et al. [22, 1] propose an interesting variation of sequential play by allowing each agent to reveal its action choice to its opponent first. If one agent chooses to reveal its action choice to its interacting partner first,[1] then its partner will make its best response accordingly. In this way, for some games, the agents are able to coordinate on POSNE outcomes under self-play, while inefficient Nash equilibria in the single-stage game would be reached without the action revelation mechanism. However, there is inadequacy in this approach since in an open environment we may not have control on the strategies of all agents. Within an open environment, the agents are usually designed by different parties and may have not the incentive to follow the strategy we design. To this end, in this work we adopt the "AI agenda" [23] by assuming that the opponent agent will not adopt the strategy we design. We assume that the opponent agent is individually rational and may adopt one of the following well-known rational strategies: Q-learning [25], WoLF-PHC [4], and Fictitious play (FP) [9] following previous work [8]. In "AI agenda" [23], one commonly adopted direction is considering how to obtain as high rewards as possible by exploiting the opponents [21]. However, we are more interested in how the opponents can be influenced towards coordination on SOSNE outcomes through repeated interactions [8].

All these above rational strategies share the common characteristic of myopic rationality, i.e., all of them follow the principle of making best responses towards their opponents based on their current estimations. One can take advantage of this characteristic when interacting with this kind of opponents. We consider an interesting variant of sequential play with decision entrustment. In addition to choose from their original action spaces, the agents are given a free option of deciding whether to entrust its opponent to make decision for itself or not (denoted as action F). If agent i asks its opponent j to make decision for itself (choosing action F), then the agents will execute the action pair assigned by agent j in this round. The motivation behind is to investigate whether the introduction of the additional action F can give the rational opponents additional incentive to coordinate towards socially optimal outcomes. Similar idea of introducing "leader" and "voter" agents has been adopted in investigating multi-agent learning in multi-agent resource selection problems [18], which has been shown to be effective for agents to coordinate on optimal utilizations of the resources.

We propose a novel learning strategy TaFSO (*T*each *a*nd *F*ollow towards *S*ocial *O*ptimality), which combines both characteristics of *teacher* and *follower* strategies. The characteristics of being a *teacher* strategy are exhibited through the implementation of sequential play and action entrustment mechanisms: a TaFSO agent rewards its opponent by choosing some socially optimal outcome as their joint action if its opponent chooses action F, and punishes its opponent otherwise. The TaFSO strategy also has the characteristic of a

[1] One agent is randomly chosen to reveal its action in case that both agents choose to reveal their actions simultaneously.

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 3	0, 5
	D	5, 0	1, 1

(a)

1's payoff, 2's payoff		Player 2's action	
		L	R
Player 1's action	U	1, 0	3, 2
	D	2, 1	4, 0

(b)

Fig. 1: Payoff matrices for (a) prisoner's dilemma game, and (b) stackelberg game

follower strategy in that it always tries to obtain as much payoff as possible (reducing the punishment cost) while punishing its opponent when its opponent does not choose action F.

A number of different learning strategies [16, 8] based on punishment mechanism have been developed to achieve more efficient outcomes instead of single-stage Nash equilibrium. Compared with previous work [16, 8], the TaFSO strategy has the following advantages. Firstly, it can be guaranteed that the opponent never perceives the wrong punishment signal and thus the coordination efficiency among agents is greatly improved. Secondly, the agent adopting the TaFSO strategy always picks an action in the best response to the opponent's strategy from the set of candidate actions suitable for punishment instead of adopting the minmax strategy, thus the punishment cost is reduced. Thirdly, due to the introduction of action F, the agents are guaranteed to always coordinate on the same optimal outcome even if multiple optimal outcomes coexist. Extensive simulations have been performed to evaluate the performance of the TaFSO strategy under self-play and also against the class of rational learners. Simulation results show that better performance under a number of evaluation criteria can be achieved compared with previous work [8, 3, 22].

The remainder of the paper is structured as follows. The related work is given in Section 2. In Section 3, the learning environment and the learning goal of SOSNE are introduced. In Section 4, we present the learning approach TaFSO under decision entrustment mechanism in the context of two-player repeated games. Experimental simulations and performance comparisons with previous work for both the case of playing against the class of rational opponents and self-play are presented in Section 5. In the last section, we conclude our paper and describe the future work.

2 Related Work

Considering the inefficiency of pursuing Nash equilibrium solution, a number of approaches [24, 17, 22, 3] have been proposed targeting at an alternative solution: Pareto-optimal solution. Some work [24, 17] focuses on the PD game only, and the learning goal is to achieve Pareto-optimal solution of mutual

cooperation instead of Nash equilibrium solution of mutual defection. There also exists some work [22, 1, 3] which addresses the problem of achieving Pareto-optimal solution in the context of general-sum games.

Sen et al. [22, 1] proposed an interesting learning mechanism of sequential play with action revelation. Under this mechanism, each agent is allowed to choose to inform the other agent of its action choice at the beginning of each round. If one agent chooses to tell its opponent its action choice beforehand, then its opponent will make its best response accordingly. Each agent adopts an expected utility based probabilistic learning strategy based on Q-learning algorithm to make its decision during each round. Simulation results show that agents using action revelation strategy under self-play can achieve Pareto-optimal outcomes which dominate Nash Equilibrium in certain games, and also the average performance with action revelation is significantly better than Nash Equilibrium solution over a large number of randomly generated game matrices. The action revelation mechanism here can be considered as a coordination signal introduced into the learning process to facilitate the coordination towards Pareto-optimal outcomes between agents. However, they only focused on the case of self-play, and also still cannot solve the problem of coordinating on mutual cooperation in the PD game. In this work, we propose an alternative coordination signal by allowing action entrustment to achieve better coordination between agents on socially optimal outcomes.

Banerjee et al. [3] proposed the conditional joint action learning strategy (CJAL) under which each agent takes into consideration the probability of an action taken by its opponent given its own action, and utilize this information to make its own decision. Simulation results show that agents adopting this strategy under self-play can learn to converge to the pareto-optimal solution of mutual cooperation in prisoner's dilemma game when the game structure satisfies certain condition. However, this strategy mainly focuses on the prisoner's dilemma game and only works when the structure of the prisoner's dilemma game satisfies certain property. Besides, this strategy is based on the assumption of self-play, and there is no guarantee of its performance against the opponents using different strategies.

A number of work [15, 16, 8] made the next step by assuming that the opponent may adopt different strategies instead of self-play. One natural way of enforcing the opponents to cooperate is to adopt the Folk theorem [19] in the literature of Game Theory. The basic idea of the Folk theorem is that there are some strategies based on the punishment mechanism which can enforce desirable outcomes and are also in Nash equilibrium of the infinitely repeated game, assuming that all players are perfectly rational. Our focus, however, is to utilize the ideas in Folk theorem to design efficient strategy against adaptive best-response opponents from the learning perspective. Besides the strategies we explore need not be in equilibrium in the strict sense, since it is very difficult to construct a strategy which is the best response to a particular learning strategy such as Q-learning. In this direction, a number of teacher strategies [15, 16] have been proposed to induce better performance from the opponents via punishment mechanisms, assuming that the opponents adopt best-response

strategies such as Q-learning. However, as mentioned in [8], there are a few disadvantages of this work that need to be addressed. The Godfather++ strategy [16] does not take the agent's own payoff into account during the punishment phase, which can make the punishment cost unnecessarily high. Besides, the agents using Godfather++ may be not able to coordinate their action choices successfully when the target solution is not unique particularly in the case of self-play.

Based on the teacher strategy Godfather++ [16], Crandall and Goodrich [8] proposed the strategy SPaM employing both teaching and following strategies simultaneously. They evaluated the performance of the SPaM strategy using a number of representative two-player games under the case of both self-play and playing against best-response learners, provided that the game structure is observable. The SPaM strategy remedies the disadvantages of Godfather++ in that the follower strategy part of the SPaM strategy enables a SPaM agent to reduce its punishment cost as much as possible when it punishes its opponent. Also it is empirically shown that the SPaM agents can always coordinate on the socially optimal solution under self-play. However, the average performance of the SPaM agent when matching against the best-response learners is not as good as its performance under self-play, and mis-coordinations occur with certain probabilities. In contrast, the TaFSO strategy we propose here learns to teach selfish opponents based on the action entrustment mechanism, which thus is able to prevent the opponents from perceiving the punishment signal by mistake.

There also exist other learning algorithms [20, 21, 6] assuming that the opponents may be adaptive and concentrating on how to achieve best-response against different types of opponents apart from the case of self-play. Two different types of opponents have been considered: stationary opponents and opponents adopting conditional strategies where their action choices depend on the most recent k periods of past history. The authors theoretically proved that their strategies can achieve ϵ-best response against the class of opponents they consider while also guarantee the maximin payoff against any other opponent and achieve Pareto-optimal Nash equilibrium of the stage game under self-play. Different from their work, we assume that the opponents are rational and focus on the most commonly adopted rational learning strategies. Besides, our goal is to achieve system-level efficiency, socially optimal outcomes, instead of achieving best-response against the opponents.

3 Learning Environment and Goal

In this paper, we focus on the class of two-player repeated normal-form games. Formally a two-player normal-form game G is a tuple $\langle N, (A_i), (u_i) \rangle$ where

- $N = \{1, 2\}$ is the set of players.
- A_i is the set of actions available to player $i \in N$.
- u_i is the utility function of each player $i \in N$, where $u_i(a_i, a_j)$ corresponds to the payoff player i receives when the joint action (a_i, a_j) is achieved.

At the end of each round, each agent receives its own payoff based on the agents' joint action and also observes the action of its opponent. Each joint action of the agents is also called an outcome of the game.[2]

Two examples of normal-form games (prisoner's dilemma game and stackelberg game) are already shown in Fig. 1a and Fig. 1b respectively.

Following the setting in [15, 8], we assume that the opponent is individually rational, and specifically we consider the opponent may adopt one of the following well-known rational strategies: Q-learning [25], WoLF-PHC [4],[3] and Fictitious play [9]. Q-learning is a rational learning algorithm that has been widely applied in multiagent interacting environments. It has been proved that the agents using Q-learning algorithm converge to some pure strategy Nash equilibrium in deterministic cooperative games only [7], but no guarantees on which Nash equilibrium that will be converged to. WoLF-PHC is empirically shown to converge to a Nash equilibrium in two-player two-action games, however, similar with Q-learning, the Nash equilibrium that the agents converge to may be extremely inefficient. Finally, fictitious play is a rational learning strategy widely studied in game theory literature, and it is guaranteed to converge to a Nash equilibrium in certain restricted classes of games (e.g., games solvable by iterated elimination of strictly dominated strategies). Under Fictitious play, each agent keeps the record of its opponent's action history, and chooses its action to maximize its own expected payoff with respect to its opponent's mixed strategy (obtained from the empirical distribution of its past action choices). For our purpose, we do not take into consideration the task of learning the game itself and assume that the game structure is known to both agents beforehand.

It is well-known that every two-player normal-form game with finite actions has at least one pure/mixed strategy Nash equilibrium [19]. Under a Nash equilibrium, each agent is making its best response to the strategy of the other agent and thus no agent has the incentive to unilaterality deviate from its current strategy.

Definition 1 A *pure strategy Nash equilibrium* for a single-shot two-player normal-form game is a pair of strategies (a_1^*, a_2^*) such that

1. $u_1(a_1^*, a_2^*) \geq u_1(a_1, a_2^*)$, $\forall a_1 \in A_1$
2. $u_2(a_1^*, a_2^*) \geq u_2(a_1^*, a_2)$, $\forall a_2 \in A_2$

If the agents are allowed to use mixed strategy, then we can naturally define the concept of *mixed strategy Nash equilibrium* similarly.

Definition 2 A *mixed strategy Nash equilibrium* for a single-shot two-player normal-form game is a pair of strategies (π_1^*, π_2^*) such that

[2] Note that in general an outcome is a profile of mixed strategies of all agents [19], and a profile of pure strategies is a special case. In this paper, we adopt the meaning that an outcome is a pure strategy profile unless otherwise mentioned.

[3] WoLF-PHC is short for Win or Learn Fast - policy hill climbing.

1. $\bar{U}_1(\pi_1^*, \pi_2^*) \geq \bar{U}_1(\pi_1, \pi_2^*)$, $\forall \pi_1 \in \Pi(A_1)$

2. $\bar{U}_2(\pi_1^*, \pi_2^*) \geq \bar{U}_2(\pi_1^*, \pi_2)$, $\forall \pi_2 \in \Pi(A_2)$

where $\bar{U}_i(\pi_1^*, \pi_2^*)$ is player i's expected payoff under the strategy profile (π_1^*, π_2^*), and $\Pi(A_i)$ is the set of probability distributions over player i's action space A_i. A mixed strategy Nash equilibrium (π_1^*, π_2^*) is degenerated to a *pure strategy Nash equilibrium* if both π_1^* and π_2^* are pure strategies.

Pure/mixed strategy Nash equilibrium in single-stage games has been commonly adopted as the learning goal to pursue in previous work [14, 25, 4, 10], however, it can be extremely inefficient in terms of the payoffs the agents receive (see the example in Fig. 1a). To this end, in this paper, we set our learning goal to converging to socially optimal outcome sustained by Nash equilibrium (SOSNE). A SOSNE outcome is an outcome maximizing the sum of all players' payoffs among all possible outcomes which correspond to a Nash equilibrium payoff profile when the game is infinitely repeated with limit of means criterion.[4] Compared with converging to Nash equilibrium in single-stage games, converging to a SOSNE outcome not only achieves system-level efficiency, but also maintains system stability since any agent deviating from the SOSNE outcome can be punished successfully which is guaranteed by the Nash folk theorem [19]. For example, consider the repeated prisoner's dilemma game in Fig. 1a. (C, C) is the only SOSNE outcome in this game, since it is the only outcome under which the sum of both agents' payoffs is maximized, and also it corresponds to a Nash equilibrium payoff profile in the limit of means infinitely repeated PD game.

To formally define the concept of SOSNE, let us define the concept of *outcome sustained by Nash equilibrium (OSNE)* first.

Definition 3 For any two-player normal-form game G, let us denote the set of Nash equilibrium payoff profiles in the corresponding infinitely repeated game under the limit of means criterion as $\mathcal{P}$. An outcome (a_1, a_2) is an *outcome sustained by Nash equilibrium (OSNE)* if and only if there exists a payoff profile $(p_1, p_2) \in \mathcal{P}$ such that $u_1(a_1, a_2) = p_1$ and $u_2(a_1, a_2) = p_2$.

In Definition 3, for an outcome to be an OSNE, we explicitly require that its payoff profile must correspond to one Nash equilibrium payoff profile in the corresponding infinitely repeated game under the limit of means criterion. From the Nash folk theorem [19], we know that for any two-player game, a Nash equilibrium payoff profile of the limit of means infinitely repeated game must be both feasible and enforceable (i.e., Pareto-dominates the minimax payoff profile). Thus for any OSNE outcome, its payoff profile must also Pareto-dominate the minimax payoff profile. In other words, the set of OSNE outcomes

[4] A preference relation $\succsim_i$ for player i is defined under the limit of means criterion if it satisfies the following property: $O_1 \succsim_i O_2$ if and only if $\lim_{t\to\infty} \Sigma_{k=1}^{t}(p_1^k - p_2^k)/t \geq 0$, where $O_1 = (a_{i,t}^1, a_{j,t}^1)_{t=1}^{\infty}$ and $O_2 = (a_{i,t}^2, a_{j,t}^2)_{t=1}^{\infty}$ are the outcomes of the infinitely repeated game, and p_1^k and p_2^k are the corresponding payoffs player i receives in round k of outcomes O_1 and O_2 respectively.

consists of all and only those outcomes whose payoff profiles Pareto-dominate the minimax payoff profile of the single-stage game. Based on the definition of OSNE, we can easily define the concept of SOSNE as follows.

Definition 4 For any two-player normal-form game G, an outcome (s_1^*, s_2^*) is a *socially optimal outcome sustained by Nash equilibrium (SOSNE)* if and only if the sum of both players' payoffs under (s_1^*, s_2^*) is the highest among all possible OSNE outcomes.

To check whether an outcome is a SOSNE outcome, we only need to find all outcomes that Pareto-dominates the minimax payoff profile of the single-stage game and examine whether the sum of both players' payoffs under this outcome is the highest among all the candidate outcomes. Taking the stacklberg game shown in Fig. 1b as an example, let us represent each payoff profile as a point in 2-dimensional space shown in Fig. 2. In this figure, the x-axis represents the payoff to the row player (player 1) and the y-axis denotes the payoff to the column player (player 2). All the feasible payoff profiles in repeated game under the limit of means criterion are within the triangle area with three vertices of (1, 0), (4, 0) and (3, 2), and also it is easy to check that the minimax payoff profile corresponds to the point (2, 1). A payoff profile is Pareto-dominated by all feasible payoff profiles that lies in the right or above it. Therefore, in this example, the set $\mathcal{P}$ of Nash equilibrium payoff profiles (i.e., the set of points Pareto-dominating the minimax payoff profile (2, 1)) is represented by the sub triangle with the three vertices: Minimax, c, and SO. Among all the outcomes whose payoff profiles are within this sub triangle, the sum of both players' payoffs are the highest under the outcome (U, R). Therefore the outcome (U, R) corresponding to the payoff profile SO is a SOSNE outcome. If there exists a SOSNE outcome in a game, it means that this outcome is enforceable since this outcome must correspond a Nash equilibrium payoff profile under the limit-of-means criterion. Therefore from a single agent's perspective, it always has the capability of enforcing any perfectly rational opponents to reach this SOSNE outcome using trigger strategy. On the other hand, it is reasonable to expect that under trigger strategy, any individually rational agent would have the incentive to coordinate on the SOSNE outcome given the threat of being punished by obtaining the worse minimax payoff otherwise in the long run. Notice that a SOSNE outcome must be a POSNE, but not vice versa, and only a POSNE outcome which maximizes the sum of both players' payoffs is a SOSNE.

4 TaFSO: A Learning Approach Towards SOSNE Outcomes

In this section, we present the learning approach TaFSO aiming at achieving SOSNE outcomes. To enable the TaFSO strategy to exert effective influence on the opponent's behavior, we consider an interesting variation of sequential play by allowing entrusting decision to others. During each round, apart from choosing an action from its original action space, every agent is also given an additional option of asking its opponent to make the decision for itself.

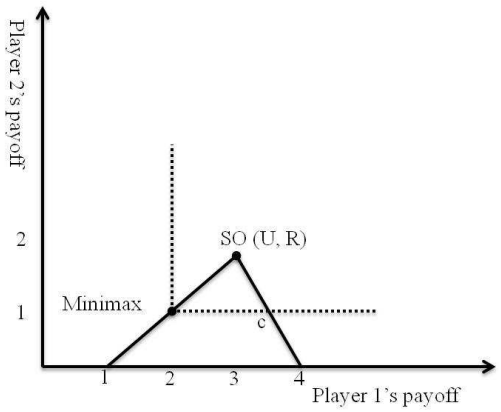

Fig. 2: Payoff profile space of the stacklberg game in Fig. 1b

The TaFSO strategy combines the properties of both *teacher* and *follower* strategies based on the action entrustment mechanism. Its *teacher* component is used to influence its opponent to cooperate and behave in the expected way based on punishment and reward mechanisms, which mainly involves the following two functions.

- If the opponent agent chooses action F, the *teacher* component will be responsible for determining which joint action to execute to reward the opponent;
- Otherwise the *teacher* component will determine the suitable set of actions for punishing the opponent for being uncooperative.

Its *follower* component is in charge of which action to choose to punish the opponent agent while ensuring that the TaFSO agent can obtain as much payoff as possible against its opponent at the same time.

4.1 *Teacher* strategy in TaFSO

During each round, apart from choosing an action from its original action space, every agent is also given an additional option of asking its opponent to make the decision for itself. Whenever the opponent j decides to entrust TaFSO agent i to make decisions (denoted as choosing action F), TaFSO agent i will select the SOSNE outcome (s_1, s_2) if there only exists one SOSNE outcome and both agents will execute their corresponding actions accordingly. If there exist multiple action pairs that are SOSNE, then the one with highest payoff for the opponent agent is selected. We assume that every agent will honestly execute the action assigned by its opponent whenever it asks its opponent to do so. If both agents choose action F simultaneously, then one of them will be randomly picked as the joint decision-maker for both agents.

We can see that the opponent j may obtain a higher payoff by deviating from action F to some action from A_j. Therefore a TaFSO agent i needs to enable its opponent j to learn to be aware that entrusting the TaFSO agent to make decisions is its best choice. To achieve this, the TaFSO agent i will teach its opponent by punishment if the opponent j chooses actions from A_j. To make it effective, the punishment must exceed the profit of deviating. In other words, the opponent j's any possible gain from its deviation has to be wiped out through one or more rounds of punishment. The TaFSO agent i keeps the record of the opponent j's accumulated gains G_j^t from deviation by each round t and updates G_j^t at the end of each round. We propose two different ways of updating the value of G_j^t based on the *forgiveness* degree of the TaFSO agent when faced with any deviation from its opponent, which are shown in Fig. 3 and Fig. 4 respectively.

- If the opponent j entrusts the TaFSO agent i to make decision for itself (i.e., $a_j^t = c$), and also its current gain $G_j^t \geq 0$, it means the opponent deviates from choosing action F and makes certain gain in previous rounds and choose not to deviate in the current round. If the TaFSO agent is nice and easy to forgive others, it will forgive the opponent and update the gain G_j^t of opponent j to zero; otherwise, it will keep the gain G_j^t of opponent j unchanged in the next round $t+1$.
- If the opponent j asks the TaFSO agent i to make decision for itself (i.e., $a_j^t = c$), and also its current gain $G_j^t < 0$, this indicates it suffers from previous deviations. In this case, if the TaFSO agent is nice and easy to forgive, it will keep the value of G_j^t unchanged; otherwise, it will update the value of G_j^{t+1} to 0 in next round.
- If the opponent j chooses its action independently and also $G_j^t > 0$, it means the opponent makes profits from previous deviations and still chooses to deviate this round. If the TaFSO agent is easy to forgive others, it will update the gain of the opponent as $G_j^t + u_j(a_i^t, a_j^t) - u_j(s_i, s_j)$; otherwise, it will update G_j^t as $G_j^{t+1} = \max\{G_j^t + u_j(a_i^t, a_j^t) - u_j(s_i, s_j), \epsilon\}$. Notice that ϵ is a small positive value which ensures that the opponent j's total gain by round $t+1$ cannot become smaller than zero since it deviates in the current round.
- If the opponent j chooses its action independently and also $G_j^t \leq 0$, it means the opponent suffers from previous deviations and still chooses to deviate this round. If the TaFSO agent is easy to forgive others, the opponent's gain is updated as $G_j^{t+1} = G_j^t + u_j(a_i^t, a_j^t) - u_j(s_i, s_j)$; otherwise, its gain is updated as $G_j^{t+1} = u_j(a_i^t, a_j^t) - u_j(s_i, s_j) + \epsilon$. That is, even though the opponent suffers from previous deviations ($G_j^t \leq 0$), it still deserves punishment by counting its previous gain as ϵ.

Based on the above updating rules of G_j^t, the TaFSO agent needs to determine which action is chosen to punish the opponent j. To do this, the TaFSO agent i keeps a teaching function $T_i^t(a)$ for each action $a \in A_i$ in each round t, indicating whether this action can be used to punish the opponent. First, the TaFSO evaluates the punishment degree $D_i^t(a)$ on the opponent by choosing

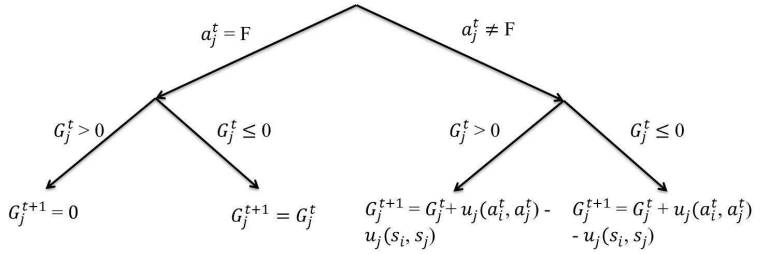

Fig. 3: The rules of calculating the opponent j's accumulated gain G_j^t by each round t (Proposal 1). Each path in the tree represents one case for calculating G_j^t.

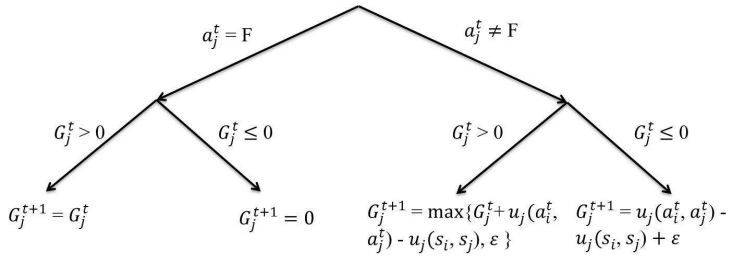

Fig. 4: The rules of calculating the opponent j's accumulated gain G_j^t by each round t (Proposal 2). Each path in the tree represents one case for calculating G_j^t.

an action a as the opponent's possible payoff loss compared with the case of following the instruction of the TaFSO agent to choose action s_2, and formally we have

$$D_i^t(a) = u_j(s_1, s_2) - E[u_j(a, b)] \tag{1}$$

where $E[u_j(a, b)]$ is the expected payoff that the TaFSO agent believes that the opponent j would obtain if the TaFSO agent chooses action a based on the past history. Formally, $E[u_j(a, b)]$ can be expressed as follows,

$$E[u_j(a, b)] = \sum_{b \in A_j} (freq_j(b) \times u_j(a, b)) \tag{2}$$

where $freq_j(b)$ is the estimated probability that the opponent j will play action b next round based on the past history (i.e., the frequency that action b was played in the past rounds).

Given an action a, if its current punishment degree is not lower than the current gain of the opponent ($D_i^t(a) \geq G_j^t$), it means this action is suitable for punishing the opponent. However, the opponent's gain may be too large to be wiped out in a single round. In this case, from the Folk theorem [19] we know that the opponent can always receive its minimax payoff $minimax_j$

when the TaFSo agent exerts punishment on it. Thus we may only expect to exert $u_j(s_1, s_2) - minimax_j$ amount of punishment on the opponent j. Overall, if $G_j^t \leq u_j(s_1, s_2) - minimax_j$, then punishing the opponent j by the amount of G_j^t is already enough; otherwise, we set our expected single-round highest amount of punishment to the value of $u_j(s_1, s_2) - minimax_j$. Therefore given an action a, the TaFSO agent evaluates whether it can be used to punish the opponent as follows,

$$T_i^t(a) = D_i^t(a) - \min\{G_j^t, u_j(s_1, s_2) - minimax_j\} \tag{3}$$

If $T_i^t(a) \geq 0$, it means it is sufficient to choose action a to punish the opponent j. The set C_i^t of candidate actions for punishment is obtained based on the values of $T_i^t(a)$. Formally we have

$$C_i^t = \{a \mid T_i^t(a) \geq 0, a \in A_i\} \tag{4}$$

We can see that C_i^t may consist of multiple candidate actions for punishing the opponent j. If $T_i^t(a) < 0$, $\forall a \in A_i$, we only choose the action with the highest $T_i^t(a)$ as the candidate action, and thus C_i^t becomes a singleton. Based on this information, the TaFSO agent i chooses an action from this set C_i^t to punish its opponent according to its *follower* strategy, which will be introduced in Section 4.2. Another point worth mentioning is that the previous two ways of updating the value of G_j^t determines the punishment degree that the TaFSO agent could exert on its opponent. The higher the punishment degree is, the less number of actions that the TaFSO agent could choose from C_i^T. Intuitively, the unforgiving version of the TaFSO agent would usually exert more harsh punishment on its opponent when the opponent deviates. Thus it is expected that the unforgiving version of the TaFSO agent can incentivize its rational opponents to coordinate on SOSNE outcomes more effectively than the forgiving version of the TaFSO agent. This hypothesis will be evaluated in the experimental part in Section 5.1.

4.2 *Follower* strategy in TaFSO

The *follower* strategy in TaFSO is used to determine the best response to the strategy of the opponent if the opponent chooses its action from its original action space. Here we adopt the Q-learning algorithm [25] as the basis of the *follower* strategy. Specifically the TaFSO agent i holds a Q-value $Q_i^t(a)$ for each action $a \in A_i \cup \{F\}$, and gradually updates its Q-value $Q_i^t(a)$ for each action a based its own payoff and action in each round. The Q-value update rule for each action a is as follows:

$$Q_i^{t+1}(a) = \begin{cases} Q_i^t(a) + \alpha_i(u_i^t(O) - Q_i^t(a)) & \text{if } a \text{ is chosen in round } t \\ Q_i^t(a) & \text{otherwise} \end{cases} \tag{5}$$

where $u_i^t(O)$ is the payoff agent i obtains in round t under current outcome O by taking action a. Besides, α_i is the learning rate of agent i, which determines

how much weight we give to the newly acquired payoff $u_i^t(O)$, as opposed to the old Q-value $Q_i^t(a)$. If $\alpha_i = 0$, agent i will learn nothing and the Q-value will be constant; if $\alpha_i = 1$, agent i will only consider the newly acquired information $u_i^t(O)$.

In each round t, the TaFSO agent i chooses its action based on the ϵ-greedy exploration mechanism as follows. With probability $1-\epsilon$, it chooses the action with the highest Q-value from the set C_i^t of candidate actions, and chooses one action randomly with probability ϵ from the original action set $A_i \cup F$. The value of ϵ controls the exploration degree during learning. It initially starts at a high value and decreased gradually to nothing as time goes on. The reason is that initially the approximations of both the teaching function and the Q-value function are inaccurate and the agent has no idea of which action is optimal, thus the value of ϵ is set to a relatively high value to allow the agent to explore potential optimal actions. After enough explorations, the exploration has to be stopped so that the agent will focus on only exploiting the action that has shown to be optimal before.

4.3 Overall Algorithm of TaFSO

The overall algorithm of TaFSO (denoted as agent i) is sketched in Algorithm 1, and it combines the *teacher* and *follower* elements we previously described. The only difference is that a special rule (line 5 to 9) is added to identify whether the opponent is adopting TaFSO or not for the case of self-play. If the opponent also adopts TaFSO, it is equivalent to the reduced case that both agents alternatively decide the joint action and thus the pre-calculated optimal outcome (s_1, s_2) is always achieved. Otherwise, during each round, the TaFSO agent first determines the optimal joint action and also the set of candidate actions based on its *teacher* strategy, and then chooses an action to execute following its *follower* strategy. The outcome of each round depends on the joint action of the TaFSO agent and its opponent, and also the Q-values and G_i^t of the TaFSO agent will be updated accordingly (Line 14 to 20).

Next we make the following observations for the TaFSO algorithm from the *teacher* and *follower* strategies' perspectives respectively and also discuss the differences with the SPaM strategy proposed in previous work [8].

4.3.1 Efficiency of the teacher *strategy*

In TaFSO, the *teacher* strategy is designed based on the entrustment mechanism we incorporate to modify the way of interaction between agents each round. The teaching goal is to let the opponent be aware that entrusting the TaFSO agent to make decisions for itself is in its best interest. The opponent is always rewarded by the payoff in the optimal outcome (s_1, s_2) when it chooses action F (not punished), and punished to wipe out its gain whenever it deviates by choosing action from its original action space based on its current gain from past deviations. In this way, it prevents the occurrence of mistaken

Algorithm 1 Overall Algorithm of TaFSO

Initialize G_i^t, $Q(a)$, $\forall a \in A_i \cup F$
Observe the game G, calculate the SOSNE outcome (s_1, s_2) with the highest payoff for the opponent.
for each round t **do**
  Compute the set C_i^t of candidate actions.
  if t = 1 **then**
    Choose action F.
  else
    if $a_j^{t-1} = F$ **then**
      Choose action F.
    else
      Choose an action a_i^t according to the *follower* strategy in Sec 4.2.
    end if
  end if
  if agent i becomes the joint decision-maker **then**
    Choose the pre-computed optimal outcome (s_1, s_2) as the joint decision.
    Update G_i^t based on the update rules in Sec 4.1.
  else
    Update $Q(a)$, $\forall a \in A_i \cup F$ following Equation 5 after receiving the reward of either outcome (a_i^t, a_j^t) or the joint action specified by its opponent.
    Update G_i^t based on the update rules in Sec 4.1.
  end if
end for

punishment and the opponent never wrongly perceives the punishment signal. Thus it is expected that for any rational agent, when it plays against a TaFSO agent, it should be able to finally learn to choose action F to maximize its individual payoff even though there may exist another outcome under which it can receive a higher payoff. One exceptional case is that when the SOSNE outcome is in the best interest of both agents, the rational opponent will not get punished by deviating from choosing action F, since the gain of the opponent by deviating from choosing action F is at most 0. Therefore, in this case, it is possible for the rational opponent to learn to choose its best action from its original action space instead of choosing action F, and the TaFSO agent will cooperate with the rational opponent to coordinate on the SOSNE outcome eventually according to the *follower* strategy.

In contrast, in previous work [8], the teaching goal is to let the opponent be aware that always choosing its corresponding action of the optimal joint action is its best choice, and the way of calculating the opponent's gain from deviation also depends on the teaching agent's own actions. The side effect is that the opponent may still be punished even when it chooses its action

from the optimal joint action, thus the opponent may misperceive the punishment signal from the teaching agent. It is expected that this would result in the teaching process less effective compared with our approach, which can be verified from the experimental results given in next section.

4.3.2 Efficiency of the follower *strategy*

From the *teacher* strategy, a set of candidate actions suitable for punishment is obtained based on the teaching function in Equation 3. Different from trigger strategy, the teaching function predicts the opponent's next-round action based on the past history, instead of assuming that the opponent always takes the maxmin strategy. This is more reasonable and efficient since the opponent does not necessarily choose the maxmin strategy and it is highly likely that there exist multiple action choices that are all sufficient to wipe out any possible gain of the opponent from past deviation.

According to the *follower* strategy, the TaFSO agent learns the relative performance of different actions (their Q-values) against the opponent. Given the set of candidate actions obtained from the teaching function, the TaFSO agent always chooses the action in its own best interest from the candidate actions through exploration and exploitation mechanism. In this way, the TaFSO agent can reduce its own punishment cost as much as possible when still guaranteeing that it is sufficient to exert punishment on its opponent. In contrast, an agent adopting trigger strategy always picks the minimax strategy to punish its opponent in a deterministic way without taking into consideration its punishment cost, which thus may make the teaching process quite inefficient. Compared with the SPaM strategy, the difference is that the SPaM strategy adopts a variant of fictitious play to determine which action to choose in order to exploit its opponent as much as possible while punishing the opponent.

5 Experiments

In this section, we perform experimental evaluations and present the experimental results in three parts. First, in Section 5.1, we evaluate and compare the learning performance of the two different versions of TaFSO strategies with different ways of updating the gain of the opponent (described in Section 4.1) adopted, when they play against best-response learners. Second, in Section 5.2 we compare the learning performance of the TaFSO strategy with the SPaM strategy [8] when playing against different best-response learners in terms of average payoff obtained under different testbeds. Last, in Section 5.3, we focus on the case of self-play and compare the performance of TaFSO strategy under self-play with previous strategies [8, 3, 22] using the testbed adopted in previous work [3] based on a number of commonly adopted evaluation criteria [3].

5.1 Punishing the opponent: forgiving or unforgiving

In Section 4.1, we have distinguished two possible ways of updating the gain of the opponent each round based on the forgiving degree of the TaFSO agent. In this section, we evaluate the learning performance of these two versions of the TaFSO strategies when playing against best-response learners. If a best-response learner becomes the joint decision-maker for both agents, we assume that it will always choose the joint action pair with the highest payoff for itself. We perform the evaluation under a larger class of games, the 57 conflicting-interest game matrices with strict ordinal payoffs. This testbed was proposed by Brams in [5], which has been widely adopted to evaluate the learning performance of different learning algorithms [3, 2]. Generally conflicting interest games are those games in which the players disagree on their most-preferred outcomes. These 57 game matrices cover all the structurally distinct two-player two-action conflicting interest games and we simply use the rank of each outcome as its payoff for each agent. All the 57 games are listed in Appendix A.

Fig. 5 shows the average payoffs of the two versions of the TaFSO learners when playing against the WOLF-PHC learner over both roles (as either the row or column player) and 100 runs across all the 57 game matrices.[5] It is interesting to notice that the unforgiving TaFSO agent (with proposal 2) actually achieves statistically significant higher average payoff than the TaFSO agent that is more forgiving (with proposal 1). We hypothesize that it is because the unforgiving version of the TaFSO agent is able to exert more effective punishment on the opponent when the opponent deviates from choosing action F, thus incentivize the opponent to switch back to choose action F more effectively. In contrast, the forgiving version of the TaFSO agent is more easy to forgive the previous deviations of the opponent, thus in some games the opponent may misperceive the punishment signal from the TaFSO agent and still pursue the maximization of its individual payoff through deviation.[6]

Fig. 6 shows the average payoffs of the WOLF-PHC learner when playing against the two different versions of the TaFSO agents over both roles (as either the row or column player) and 100 runs across all the 57 game matrices. We can see that the WOLF-PHC learner is able to receive statistically significant higher average payoff when playing against the forgiving version of the TaFSO agent (with proposal 1). This is consistent with the results of the TaFSO agents and can be explained in a similar way. When playing against the forgiving version of the TaFSO agent, since the TaFSO agent is easy to forgive the previous deviations of the opponent, it is more likely for the WOLF-PHC learner to deviate from choosing action F to maximize its individual payoffs.

[5] Similar results can be observed when the opponent adopts other types of best-response strategies (Q-learning and FP) and are omitted here.

[6] Note that theoretically, the opponents should be able to understand the punishment signal given enough explorations. In practice, due to the exploration schedule, the opponents do not explore enough to understand the punishment and thus settle on a sub-optimal strategy.

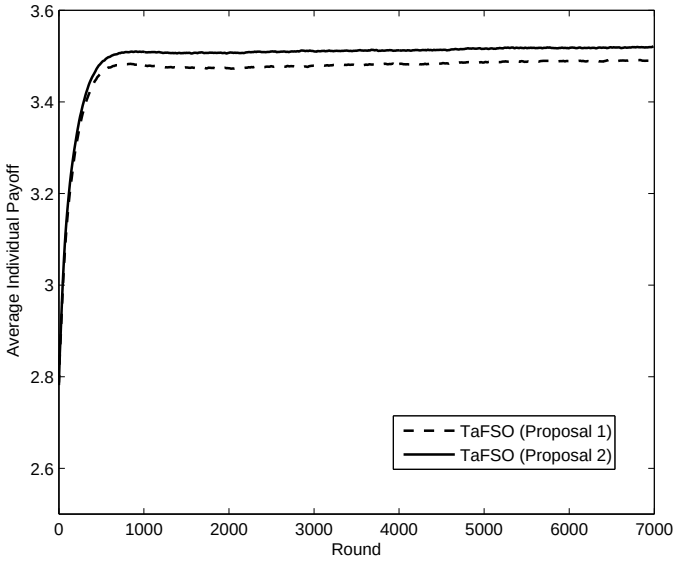

Fig. 5: Average payoffs of the two different versions of the TaFSO agents with different proposals of updating the gain of the opponent adopted

From previous results, we know that the unforgiving TaFSO agent can achieve higher individual payoff than the forgiving TaFSO agent at the cost of reducing the payoff of its opponent (the WOLF-PHC learner). Next we compare the utilitarian social welfare of agents when these two different versions of the TaFSO agents play against the WOLF-PHC learner over both roles (as either the row or column player) and 100 runs across all the 57 game matrices, which is shown in Fig. 7. We can observe that the unforgiving TaFSO agent (with proposal 2) is able to achieve statistically significant slightly higher utilitarian social welfare than the TaFSO agent who is more forgiving (with proposal 1). Intuitively, the TaFSO agent which is more forgiving (with proposal 1) makes more concessions to the WOLF-PHC agent, which thus allows the WOLF-PHC agent to obtain higher individual payoffs. However, those outcomes under which the WOLF-PHC agent can obtain higher payoff are usually not socially optimal, thus the side effect of this kind of forgiveness and concession is the decrease of the group's overall utility.

5.2 Against Best-response Learners

In this part, we evaluate the performance of TaFSO strategy against the opponents adopting a variety of different best-response strategies. From Section 5.1, we have known that the learning performance of the unforgiving version

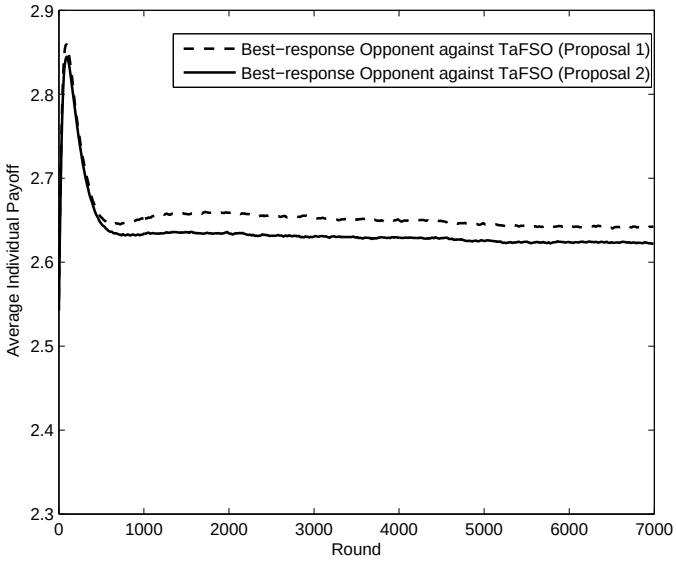

Fig. 6: Average payoffs of the WOLF-PHC learner when playing against two different versions of the TaFSO agents

of the TaFSO agent is better and thus this version of the TaFSO agent will be used for all the following experimental evaluations unless mentioned otherwise. For the best-response opponents, we assume that the opponent may adopt one of the following best-response strategies: Q-learning [25], WoLF-PHC [4], and Fictitious play (FP) [9]. Similar to the previous section, here we assume that an agent using either of the previous best-response strategies will always choose the joint action pair with the highest payoff for itself when it becomes the decision-maker for both agents. We compare the performance of TaFSO with SPaM [8] against the same set of best-response opponents under different testbeds. The first set of testbed we adopt here is the same as the one in previous work [8] by using the following three representative games: prisoner's dilemma game (Fig. 1a), game of chicken (Fig. 8a), and tricky game (Fig. 8b).

For the prisoner's dilemma game, the socially optimal and also SOSNE outcome is (C, C) , in which both agents receive a payoff of 3. For the game of chicken, the target solution is also (C, C), in which both agents obtain a payoff of 4. In the tricky game, the socially optimal and also SOSNE outcome is (C, D), and the agents' average payoffs are 2.5. Table 1 shows both the agents' utilitarian social welfare when both the TaFSO and SPaM strategies are adopted to repeatedly play the above representative games against

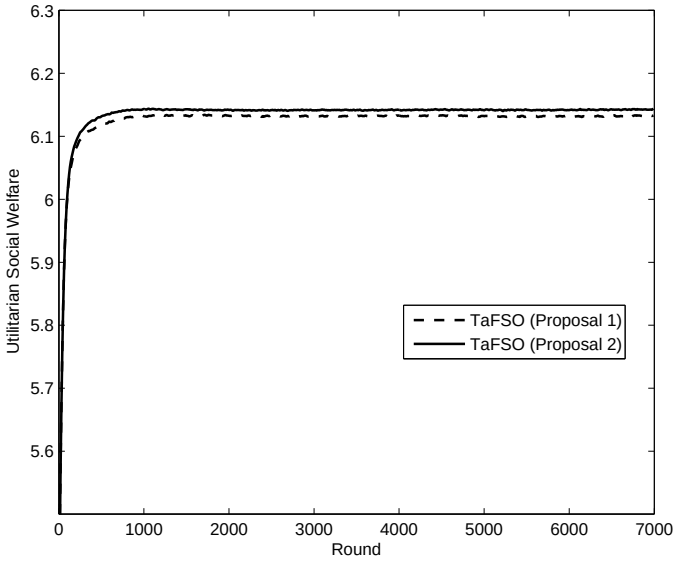

Fig. 7: Utilitarian social welfare of the two different versions of the TaFSO agents with different proposals of updating the gain of the opponent adopted

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	4, 4	2, 5
	D	5, 2	0, 0

(a)

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	0, 3	3, 2
	D	1, 0	2, 1

(b)

Fig. 8: Payoff matrices for (a) game of chicken, and (b)tricky game

different rational opponents.[7] We can see that when the SPaM strategy is adopted, the agents can always receive the utilitarian social welfare corresponding to the socially optimal outcomes for different games. In contrast, for those cases when the SPaM strategy is adopted, the agents' utilitarian social welfare is relatively lower than the sum of the payoffs under the socially optimal outcomes. The main reason is that the opponent agents adopting the best-response strategies may wrongly perceive the punishment signals from

[7] Note that only the payoffs obtained after 500 rounds are counted here since at the beginning the agents may achieve very low payoffs due to initial explorations. The results are averaged over 50 runs. For the tricky game, the payoffs of both TaFSO and SPaM learners are averaged over the cases when they play as the row or column players.

Table 1: The agents' utilitarian social welfare when both the TaFSO and SPaM strategies play against a number of best response learners in three representative games

Utilitarian Social Welfare	Prisoner's Dilemma Game	Game of Chicken	Tricky Game
TaFSO vs. Q-learning	6.0	8.0	5.0
SPaM vs. Q-learning	5.23	7.34	4.56
TaFSO vs. WoLF-PHC	6.0	8.0	5.0
SPaM vs. WoLF-PHC	5.2	7.4	4.5
TaFSO vs. FP	6.0	8.0	5.0
SPaM vs. FP	5.4	7.82	4.68

the SPaM agent, and thus result in mis-coordination occasionally. This kind of occasional mis-coordination on the socially optimal outcomes directly results in the consequence that the utilitarian social welfare when employing the SPaM strategy is lower than that when the TaFSO strategy is adopted.

To further validate our results, next we further evaluate the performance of TaFSO strategy under a larger class of games, the 57 conflicting-interest game matrices with strict ordinal payoffs as previously mentioned. These 57 game matrices cover all the structurally distinct two-player two-action conflicting interest games. For the non-conflicting interest (common interest) games, it is trivial since there always exists a Nash equilibrium that both players prefer most and also is optimal for both players.

For these 57 conflicting-interest games, first we notice that the TaFSO learner is able to successfully incentive its rational opponent to converge to SOSNE outcome for all the three rational strategies we consider. Table 2 shows the utilitarian social welfare when the TaFSO and SPaM learners play against different rational opponents over all the 53 conflicting-interest games and both roles (the row and column players). We can see that the TaFSO learner is able to achieve higher utilitarian social welfare than the SPaM learner when playing against all the three rational opponents. We hypothesize that it is due to the rational opponent's misperception of the punishment signal from the SPaM agent in some games, which thus leads to the convergence of non-socially optimal outcomes. Taking game 32 for example, the SPaM agent expects to coordinate on the outcome (D, D), while the rational opponent may deviate by choosing action C to increase its payoff. In this case, the SPaM agent would choose action C to punish the opponent, which would reinforce the opponent to choose action C, since choosing action D is worse for itself when the SPaM agent chooses action C (performing punishment). Therefore the agents finally converge to the non-socially optimal outcome (C, C), which results in the decrease of the overall group's utility. Another example is considering game 48, in this game any rational opponent as the row player has the incentive to deviate from the optimal outcome (D, D) to increase its individual payoff, and the SPaM agent (as the column player) would choose action C to punish its

Table 2: Utilitarian social welfare under the 57 conflicting-interest games when the TaFSO and SPaM agent play against a number of rational opponents

Utilitarian social welfare	the TaFSO agent	the SPaM agent
Against Q-learning	6.275	6.05
Against WoLF-PHC	6.156	4.26
Against FP	6.21	5.73

row opponent. The punishment signal from the SPaM agent will reinforce the rational opponent to stay there: when the column agent chooses action C, the best response for the row agent is choosing action C. Accordingly, the agents will eventually converge to the inefficient non-socially optimal outcome (C, C) instead of the socially optimal outcome (D, D). In contrast, there is no misperception of the punishment signal under the action entrustment mechanism, and the TaFSO agent can successfully coordinate on the socially optimal outcomes under game 32 or 48 when playing against all rational opponents we consider.

Overall, we can see that under the action entrustment mechanism, compared with the SPaM agent, the TaFSO agent is able to induce rational opponents to converge to (socially) optimal outcomes under more percentages of games, and thus obtain higher utilitarian social welfare on average.

5.3 Under Self-play

In this section we compare the performance of TaFSO with SPaM [8], CJAL [3], Action Revelation [22] and WOLF-PHC [4] in two-player's games under self-play. Both players play each game repeatedly for 2000 time steps with learning rate of 0.6. The exploration rate starts at 0.3 and gradually decreases by 0.0002 each time step. For all previous strategies the same parameter settings as those in their original papers are adopted.

Here we again use the 57 conflicting-interest game matrices with strict ordinal payoffs proposed by Brams in [5] as the testbed for evaluation. For the non-conflicting interest games, it is trivial since there always exists a Nash equilibrium in the single stage game that both players prefer most and also is optimal for both agents. It is easy for the agents to learn to converge to this optimal Nash equilibrium for all the learning strategies we consider here and receive the maximum payoffs for both agents, and thus this type of games is not considered.

The performance of each strategy is evaluated in self-play on these 57 conflicting interest games, and we compare their performance based on the following two criteria [3]. The comparison results are obtained by averaging over 50 runs across all the 57 conflicting interest games.

Utilitarian Social Welfare The utilitarian collective utility function $sw_U(P)$ for calculating utilitarian social welfare is defined as $sw_U(P) = \sum_i^n p_i$,

where $P = \{p_i\}_i^n$ and p_i is the actual payoff agent i obtains when the outcome is converged. Utilitarian social welfare can be used as the criterion for evaluating the learning performance of each strategy under self-play in terms of average payoffs obtained when the influence of different roles (as row or column player) is taken into account.

Nash Social Welfare Nash social welfare is also an important evaluation metrics in that it strikes a balance between maximizing utilitarian social welfare and achieving fairness. Its corresponding utility function $sw_N(P)$ is defined as $sw_N(P) = \prod_i^n p_i$, where $P = \{p_i\}_i^n$ and p_i is the actual payoff agent i obtains when the outcome is converged. One one hand, Nash social welfare reflects utilitarian social welfare. If any individual agent's payoff decreases, the Nash social welfare also decreases. On the other hand, it also reflects the fairness degree between individual agents. If the total payoffs is a constant, then Nash social welfare is maximized only if the payoffs is shared equally among agents.

The comparison results based on these two criteria are shown in Table 3. We can see that TaFSO outperforms all the other four strategies in terms of the above criteria. Players using the ToFSO strategy can obtain utilitarian social welfare of 6.45 and Nash social welfare of 10.08, which are higher than all the other strategies. We also provide the average Nash equilibrium payoffs for all the 57 games for comparison purpose, which clearly shows that pursuing the goal of Nash equilibrium is less efficient compared with the goal of SOSNE. Note that the performance of WOLF-PHC approach is the worst since this approach is specifically designed for achieving Nash equilibrium in single-stage game only. For SPaM, it is possible for both SPaM learners to misperceive that their opponents are deliberately deviating from the optimal solution to increase their individual payoffs and punish their opponents simultaneously. This kind of mutual punishment can lead to punishment deadlock, which corresponds to the case that both SPaM learners always play its minmax strategy, thus both agents receive suboptimal payoffs.

For Action Revelation and CJAL, they both fail in certain types of games, e.g., the prisoner's dilemma game. For Action Revelation, self-interested agent can always exploit the action revelation mechanism and have the incentive to choose defection D, thus leading the outcome to converge to mutual defection; for CJAL, it requires the agents to randomly explore for a finite number of rounds N first and the probability of converging to mutual cooperation tends to 1 only if the value of N approaches infinity. Besides, it only works when the payoff structure of the prisoner's dilemma game satisfies certain condition [3]. For example, consider the two different versions of the prisoner's dilemma game in Fig. 9a and Fig. 9b. For both WOLF-PHC and Action Revelation, the agents always converge to the pure strategy Nash equilibrium (D, D); for CJAL, the agents can successfully learn to converge to the socially optimal outcome (C, C) for the first prisoner's dilemma game while fail to converge to (C, C) for the second one [3]. In contrast, the agents using the TaFSO strategy can always coordinate on the socially optimal outcome (C, C) for both instances of the prisoner's dilemma games under self-play.

Table 3: Performance comparison with CJAL, action revelation and WOLF-PHC using the testbed in [5]

	Utilitarian Social Welfare	Nash Social Welfare
TaFSO (our strategy)	6.45	10.08
SPaM [8]	6.10	9.25
CJAL [3]	6.14	9.25
Action Revelation [22]	6.17	9.30
WOLF-PHC [4]	6.03	9.01
Nash	6.05	9.04

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 3	0, 5
	D	5, 0	1, 1

(a)

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 3	0, 5
	D	5, 0	2, 2

(b)

Fig. 9: Payoff matrices for prisoner's dilemma game (a) version 1, and (b) version 2

6 Conclusion and Future Work

In this paper, we propose a learning strategy TaFSO consisting of both *teacher* and *follower* strategies' characteristics to achieve socially optimal outcomes. We consider an interesting variation of sequential play by introducing an additional action F for each agent. The introduction of action F serves as an additional signal to facilitate the coordinate between agents, and the adoption of this signal is voluntary and determined by the agents themselves independently. Simulation results show that a TaFSO agent can effectively influence a number of rational opponents towards SOSNE outcomes and better performance in terms of higher average payoff can be achieved compared with previous work under both the case of against a class of rational learners and self-play.

In this work, we focus on learning towards socially optimal outcomes in the cases of self-play and against the class of rational learners only. It remains unexplored on how to better utilize the characteristics of their strategies towards the optimal goals when interacting with other types of opponents. Besides, here we focus on achieving socially optimal outcomes as our targeted goal and it is sufficient to achieve one of them if there exist multiple socially optimal outcomes. One parallel direction is to investigate how to design the strategy

when the targeted solution consists of a sequence of outcomes such as achieving fairness [12].

Acknowledgements

The work described in this paper was partially supported by a CUHK Direct Grant for Research (Project ID 4055024).

References

1. Airiau S, Sen S (2006) Learning to commit in repeated games. In: AAMAS'06, pp 1263 – 1265
2. Airiau S, Sen S (2007) Evolutionary tournament-based comparison of learning and non-learning algorithms for iterated games. Journal of Artificial Societies and Social Simulation 10
3. Banerjee D, Sen S (2007) Reaching pareto optimality in prisoner's dilemma using conditional joint action learning. AAMAS'07 pp 91–108
4. Bowling MH, Veloso MM (2003) Multiagent learning using a variable learning rate. Artificial Intelligence pp 215–250
5. Brams SJ (1994) Theory of Moves. Cambridge University Press, Cambridge, UK
6. Chakraborty D, Stone P (2013) Multiagent learning in the presence of memory-bounded agents. Autonomous Agents and Multi-Agent Systems pp 1–32
7. Claus C, Boutilier C (1998) The dynamics of reinforcement learning in cooperative multiagent systems. In: AAAI'98, pp 746–752
8. Crandall JW, Goodrich MA (2005) Learning to teach and follow in repeated games. In: AAAI Workshop on Multiagent Learning
9. Fudenberg D, Levine DK (1998) The Theory of Learning in Games. MIT Press
10. Hu J, Wellman M (1998) Multiagent reinforcement learning: Theoretical framework and an algorithm. In: Proceedings of the Fifteenth International Conference on Machine Learning, pp 242–250
11. Jafari A, Greenwald A, Gondek D, Ercal G (2001) On no-regret learning, fictitious play, and nash equilibrium. In: ICML'01, pp 226–233
12. Jong S, Tuyls K, Verbeeck K (2008) Artificial agents learning human fairness. In: AAMAS'08, ACM Press, pp 863–870
13. Lauer M, Rienmiller M (2000) An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In: ICML'00, pp 535–542
14. Littman M (1994) Markov games as a framework for multi-agent reinforcement learning. In: Proceedings of the 11th international conference on machine learning, pp 322–328
15. Littman ML, Stone P (2001) Leading best-response strategies in repeated games. In: IJCAI Workshop on Economic Agents, Models, and Mechanisms

16. Littman ML, Stone P (2005) a polynomial time nash equilibrium algorithm for repeated games. Decision Support Systems 39:55–66
17. Moriyama K (2008) Learning-rate adjusting q-learning for prisoner's dilemma games. In: WI-IAT '08, pp 322–325
18. Oh J, Smith SF (2008) A few good agents: multi-agent social learning. In: AAMAS'08, pp 339–346
19. Osborne MJ, Rubinstein A (1994) A Course in Game Theory. MIT Press, Cambridge
20. Powers R, Shoham Y (2004) New criteria and a new algorithm for learning in multi-agent systems. NIPS'04 17:1089–1096
21. Powers R, Shoham Y (2005) Learning against opponents with bounded memory. In: IJCAI'05, pp 817–822
22. Sen S, Airiau S, Mukherjee R (2003) Towards a pareto-optimal solution in general-sum games. In: AAMAS'03, pp 153–160
23. Shoham Y, Powers R, Grenager T (2007) If multi-agent learning is the answer, what is the question? Artificial Intelligence 171:365–377
24. Stimpson JL, Goodrich MA, Walters LC (2001) Satisficing and learning cooperation in the prisoner's dilemma. In: IJCAI'01, pp 535–540
25. Watkins CJCH, Dayan PD (1992) Q-learning. Machine Learning pp 279–292

A Appendix

The 57 structurally distinct games mentioned in Section 5 are listed as follows.

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 4	4, 2
	D	2, 3	1, 1

(a) game 1

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 4	4, 2
	D	1, 3	2, 1

(b) game 2

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 4	4, 1
	D	2, 3	1, 2

(c) game 3

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 4	4, 1
	D	1, 3	2, 2

(d) game 4

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 4	4, 2
	D	1, 3	3, 1

(e) game 5

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 4	4, 1
	D	1, 3	3, 2

(f) game 6

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 3	4, 2
	D	2, 4	1, 1

(g) game 7

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 3	4, 2
	D	1, 4	2, 1

(h) game 8

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 3	4, 1
	D	1, 4	2, 2

(i) game 9

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 3	4, 2
	D	1, 4	3, 1

(j) game 10

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 3	4, 1
	D	1, 4	3, 2

(k) game 11

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 4	4, 1
	D	2, 2	1, 3

(l) game 12

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 4	4, 1
	D	1, 2	2, 3

(m) game 13

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 4	2, 2
	D	1, 3	4, 1

(n) game 14

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 4	2, 1
	D	1, 3	4, 2

(o) game 15

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 4	1, 2
	D	2, 3	4, 1

(p) game 16

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 4	1, 1
	D	2, 3	4, 2

(q) game 17

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 4	3, 2
	D	1, 3	4, 1

(r) game 18

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 4	3, 1
	D	1, 3	4, 2

(a) game 19

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 4	2, 3
	D	1, 2	4, 1

(b) game 20

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 4	1, 3
	D	2, 2	4, 1

(c) game 21

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 4	3, 3
	D	1, 2	4, 1

(d) game 22

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 3	4, 1
	D	2, 2	1, 4

(e) game 23

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 3	4, 1
	D	1, 2	2, 4

(f) game 24

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 2	4, 1
	D	2, 3	1, 4

(g) game 25

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 2	4, 1
	D	1, 3	2, 4

(h) game 26

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 3	4, 1
	D	1, 2	3, 4

(i) game 27

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 2	4, 1
	D	1, 3	3, 4

(j) game 28

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 2	2, 1
	D	4, 3	1, 4

(k) game 29

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 2	4, 1
	D	3, 3	1, 4

(l) game 30

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 2	3, 1
	D	4, 3	1, 4

(m) game 31

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 2	4, 1
	D	1, 4	3, 3

(n) game 32

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 4	4, 3
	D	1, 2	2, 1

(o) game 33

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 4	4, 3
	D	2, 2	1, 1

(p) game 34

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 4	4, 3
	D	1, 2	3, 1

(q) game 35

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 4	4, 3
	D	2, 1	1, 2

(r) game 36

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 4	4, 3
	D	1, 2	2, 1

(a) game 37

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 4	4, 2
	D	2, 1	1, 3

(b) game 38

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 4	4, 2
	D	1, 1	2, 3

(c) game 39

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 3	4, 2
	D	2, 1	1, 4

(d) game 40

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 3	4, 2
	D	1, 1	2, 4

(e) game 41

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 4	4, 1
	D	3, 2	1, 3

(f) game 42

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 4	3, 1
	D	4, 2	1, 3

(g) game 43

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 3	4, 1
	D	3, 2	1, 4

(h) game 44

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 3	3, 1
	D	4, 2	1, 4

(i) game 45

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 4	2, 1
	D	4, 2	1, 3

(j) game 46

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 3	2, 1
	D	4, 2	1, 4

(k) game 47

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 3	4, 2
	D	1, 1	3, 4

(l) game 48

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 4	4, 1
	D	1, 2	3, 3

(m) game 49

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 4	4, 3
	D	1, 1	3, 2

(n) game 50

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 4	2, 1
	D	1, 2	4, 3

(o) game 51

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 4	3, 1
	D	1, 2	4, 3

(p) game 52

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 3	3, 4
	D	4, 2	1, 1

(q) game 53

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 2	3, 4
	D	4, 3	1, 1

(r) game 54

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 2	4, 3
	D	3, 4	1, 1

(s) game 55

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	2, 4	4, 2
	D	1, 1	3, 3

(t) game 56

1's payoff, 2's payoff		Player 2's action	
		C	D
Player 1's action	C	3, 3	2, 4
	D	4, 2	1, 1

(u) game 57