Action Trading for Self-Interested Multi-Agent Reinforcement Learning in an Escape Room Setting

Arnold Unterauer
Ludwig Maximilian University of Munich
Munich, Germany

ABSTRACT

Multi agent systems have always successfully provided a platform to solve problems. Using reinforcement learning, agents can learn to understand their environment and thus deal with complex problems through optimisation. However, there is one major challenge: agents always try to maximize their own reward, which means that they do not take other entities into consideration and as a result act selfish. This raises the question of how to overcome this problem and enable cooperation between agents. One approach to create cooperation is action trading, which allows agents to trade with each other. This concept has already been demonstrated in a few domains, namely the Iterated Matrix Game and the Coin Game. In this work we will try to expand the approach to new environments and to take a more in-depth analysis of the trading amount. As a result, we should gain more insight into the cooperative behavior of agents and possibly improve the trading mechanism.

KEYWORDS

reinforcement learning; cooperation; multi-agent; action trading

ACM Reference Format:
Arnold Unterauer. 2021. Action Trading for Self-Interested Multi-Agent Reinforcement Learning in an Escape Room Setting. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), London, UK, May 3–7, 2021*, IFAAMAS, 6 pages.

1 INTRODUCTION

Reinforcement Learning (*RL*) [8] has become more and more popular in recent years as it can be used in all sorts of applications. With *RL* an agent is able to learn to achieve his objectives in a domain. To do this, the agent has to interact with the surroundings and use their actions to maximize their reward. The agent learns through various interactions with the environment and explores his possibilities until he learns and exploits an optimal policy π^*. If this single-agent environment is extended to an multi-agent one, the ability to achieve the goal may be affected by other agents. For example, the change in the environment no longer depends on the interaction of one agent alone but of several, which may result in an unpredictable environment for the learner. Moreover, the objectives may be dependent on other agents, which makes it impossible for the agents to solve them alone. Because agents always want to maximize their own reward due to *RL* [8], situations can arise where agents block potential rewards for other agents. This could even lead to agents failing to achieve their goals in the domain and only because of the selfish behavior of one agent. This raises the question of how selfishness can be overcome and how cooperation between agents can be established to solve problems together. This subject is studied in the field of *Game Theory* [2], which focuses on modelling mathematical decision situations, called *Games*, in order to derive rational decision-making of entities. A distinction must also be made between cooperative and non-cooperative games, see [2]. For a game to be considered cooperative, it is necessary that *binding agreements* can be established. Binding agreements can be enabled by communication, for example. This has also been addressed in recent papers [4], [3], [1], where various concepts have been introduced. One of these approaches is *Action Trading* [7], which allows agents to exchange their reward for an action from another agent. The *Action Trading* approach can be used to turn a non-cooperative environment into a cooperative one. As a result, agents can increase not only their own reward, but also the overall reward, as shown in [7] using the *Iterated Matrix Game* and the *Coin Game* environments. For our work, we will extend the concept of trading agents to the new environments *Escape Room* and *Smart Factory*, and investigate more closely how to compensate the performed action. Since the compensation in [7] was selected by hand, we want to find a method that allows us to calculate it automatically.

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, London, UK.

2 RELATED WORK

There are many approaches to implement cooperation in *multi-agent systems*. A recent concept, called *Action Trading* [7], enables agents to trade reward for the actions of others. The action space of the agents is expanded for this purpose, which enables the agents to make offers to others. After an agent has received an offer, he can then decide whether or not to follow the proposed action. By following the offer the agent will be compensated by the proposer with reward. This trading mechanism was already shown successful in the domains *Iterated Matrix Game* and the *Coin Game*. However, there are still open questions, such as the compensation amount for the trade. In [7] the agents were paid with a fixed compensation amount, which was handmade for the environments. Consequently, each time the *Action Trading* concept is applied, the user has to analyze the environment and determine an optimal compensation amount. However, since every agent values a trade depending on the situation, we cannot generalize that there is a fixed reward for the compensation. In this work, we want to investigate the cooperative behavior of agents by using a variety of compensation calculations and find a way to automate compensation. Furthermore we use the *Action Trading* approach to introduce it to new environments, *Escape Room* and *Smart Factory*, and validate the cooperation between agents. The *Escape Room* domain was recently introduced in [9], where agents learned to incentivize others to cooperate. This environment allows to investigate the cooperation

between entities as it can only be solved by cooperating. In the *Escape Room* two agents are trapped in a room and have to pull a lever to open a door, which allows the escape from the room. However, only one agent is able to pull it and is punished for doing so. As he is is not interested in leaving the room, the other agent has to motivate him to pull the lever. Since the pulling agent is trying to maximize his reward, he will not pull the lever, which would result in punishment and no benefit to the agent. Therefore non-cooperative agents will be trapped in the room and will never exit the environment. We want to extend the *Escape Room* of [9] into a 2D grid world. This allows us to examine the cooperation between agents with regard to long-term planning, since the pulling agent must first be encouraged to approach the lever and then pull it.

3 ENVIRONMENTS

Multi-agent environments can be used to investigate different cooperative game theory aspects. We therefore introduce the environments *Escape Room* and *Smart Factory*. In the *Escape Room* domain, agents have to escape from a room. This can only be achieved by opening the door with a lever, which can only be pulled by one specific agent. There is a hurdle, because the agent pulling the lever has to be motivated first. In the *Smart Factory*, agents have to perform a number of tasks that require them to process machines. However, these machines are limited, which creates competition. Additionally there are different priorities to create asymmetrical rewards, which leads to agents with a greater need to process machines first.

3.1 Escape Room

Figure 1: Escape Room: two agents (blue and purple) are trapped in a room. In order to escape, one specific agent has to pull the lever (red). But since the agent will be punished for this, both agents will end up in captivity.

The problem of selfish agents can be observed in multiple environments, where agents rather maximize their own reward than cooperating with each other. The *Escape Room* explicitly addresses the hurdle to cooperation, where two agents are trapped in a room and have to work together to escape from it. An agent has to pull a lever that opens the door and thus allows to leave the room. But only one of the two agents can perform this task. Moreover, the asymmetrical distribution of the rewards leads to two agents with different attitudes. The first agent (agent 0) does not want to stay in the room, because he is punished for every step he takes in the environment. However, when he reaches the door he ends the episode and receives a large amount of reward. On the other hand, there is the second agent (agent 1) who can pull the lever. He doesn't care if he is trapped in the room or not, because he doesn't lose any reward per step. However, he doesn't like to change the environment, so he gets a penalty when he pulls the lever 6. This conflict of interest leads to an unsatisfactory situation for agent 0, because agent 1 learns not to pull the lever and therefore agent 0 cannot leave the room. In order for agent 0 to achieve his goal of leaving the room, it is necessary that both agents cooperate and agent 1 have to be motivated to pull the lever. This is where the selfish behaviour of agents fall short, as they are only interested in their own objective. As a result the *Escape Room* environment can be used to test and compare different cooperative approaches to motivate agent 1.

3.2 Smart Factory

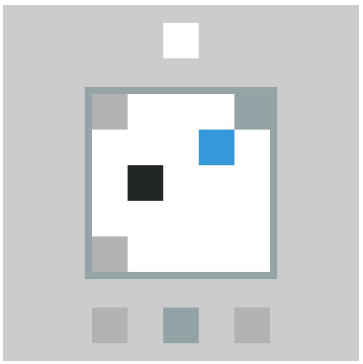


Figure 2: Smart Factory: two agents (black and blue) have to process machines to complete tasks, shown at the bottom. However, they have different priorities, as indicated above, and must therefore learn to assign the machines accordingly.

Another environment to investigate social behaviour of agents is the *Smart Factory*. There agents have to complete a given amount of tasks, where they have to process machines. These machines have different types and an agent is only able to process machines, which correspond to the next task. For each completed task the agent is rewarded, but he is also penalized for each step he takes in the environment. This penalty should motivate the agent to complete the tasks as fast as possible, as the agents want to maximize their rewards. However, the machines in the environment are a limited resource and the agents have to complete with each other to process the machines first. Additionally the agents are given different priorities, which influences the reward and the penalty of each agent. This results in an asymmetrical reward distribution, which leads to agents who have a greater need for processing machines. But as agents always want to maximize their own rewards, this aspect of asymmetrical reward distribution is not recognized by non-cooperative agents. This problem can be solved by cooperative agents, who will respect the different rewards and penalties.

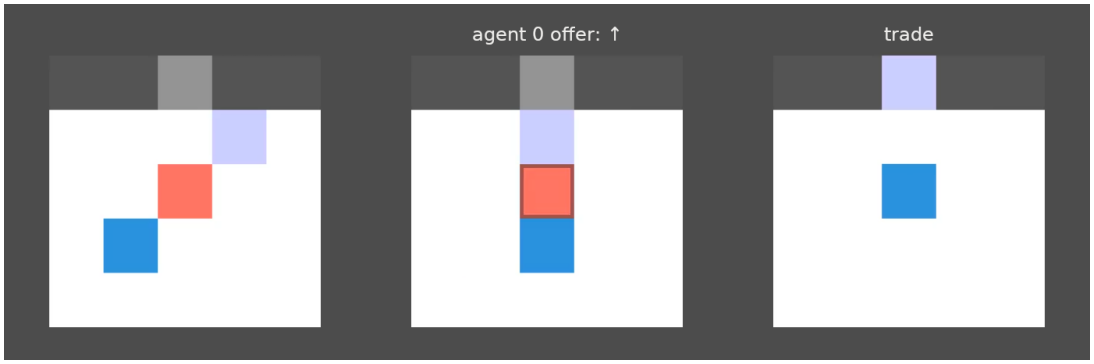


Figure 3: Trading Agents: One agent (purple) makes an offer to the other agent (blue) based on his previous observations. The recipient (blue) can either choose to follow the offer, indicated by the frame of the cell, or reject it. By following the offer, the agent is compensated by the proposer (purple) and the trade is therefore successful.

4 ACTION TRADING

Most multi-agent environments in which agents operate are of a selfish nature which do not support cooperation between agents. *Action trading* is a mechanism that allows a non-cooperative environment to be transformed into a cooperative one. The trading approach enables agents to trade their reward for the following action of another agent. For this purpose, the original actions are extended by offer actions, which allow agents to propose an action to others. This results in $|A_M + A_M^2|$ actions for each trading agent, where A_M are the original actions without offers and A_M^2 are the actions with offers. In the following we will refer to the agents who make an offer as *proposers* and the agents who receive an offer as *receivers*. Whenever an agent makes an offer, the outgoing and incoming offers are displayed in the observations of both the proposer and the receiver. Based on these observations agents have to choose their actions at the next time step. If the offer and action match, the receiver is then paid for his performed action by the proposer. The amount of compensation C the agent receives can be calculated as follows: Every agent wants to maximize their own reward and they therefore always choose the most promising action available to him. Hence the receiver has to be compensated for taking an action with less expected reward. The expected rewards $Q^\pi(s, a)$ under the policy π for an action a in state s is given by:

$$Q^\pi(s, a) = E[\sum_{t=0}^{\infty} \gamma^t R_{t+1} | s, a] \tag{1}$$

Where the future rewards are discounted by the factor γ. To make two actions a and a' indifferent in regard to the expected reward, $Q^\pi(s, a)$ and $Q^\pi(s, a')$ have to be the same. For the receiver to choose the proposed action a' over the most promising action a_{max}, the expected reward has to be equal or higher. Therefore the compensation C has to meet the requirement of:

$$Q^\pi(s, a') + C \geq Q^\pi(s, a_{max}) \tag{2}$$

With the bellman equation, the expected reward of an action a can also be denoted in the following way [6]:

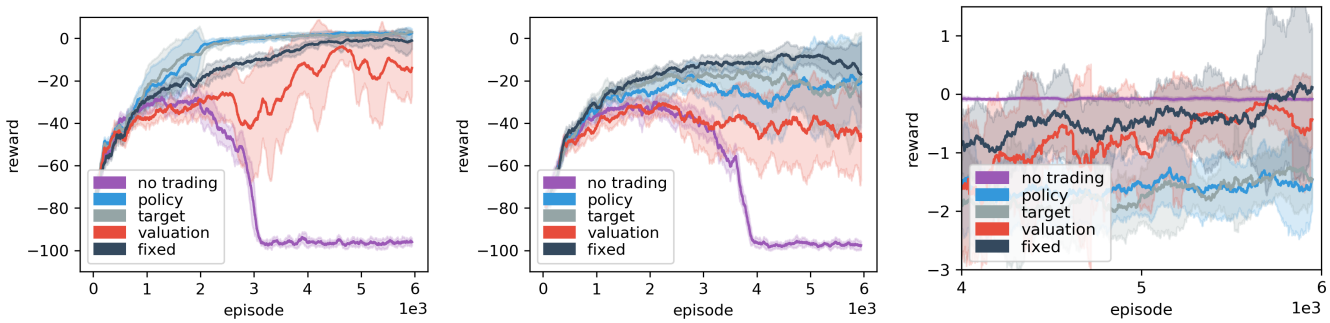
Algorithm 1: Action Trading

```
t ← 0
for each time step t do
    for each agent in agents do
        m(t), o(t) ← agent selects an action a consisting of
          movement m(t) and offer o(t)
        Move agent in environment with m(t)
        if m(t) matches o(t − 1) of other agent then
            Other agent pays agent with compensation C
        end
    end
    t ← t + 1
end
```

$Q^\pi(s_0, a_0) = R(s_0, a_0) + \gamma R(s_1, \pi_1) + ... + \gamma^t R(s_{t+1}, \pi_{t+1})$, where the future rewards R at time step t are discounted by γ. Since the compensation payment of the trading agents takes place one time step later, the compensation must be adjusted with the factor γ. To make the proposed action more appealing, we can multiple the compensation with a mark up M_c which leads to the total compensation of:

$$\text{Compensation: C} = M_c \frac{Q_{max}(t) - Q_{offer}(t)}{\gamma} \tag{3}$$

Since the proposer and the receiver want to maximize their individual reward, the action trade has to be beneficial to both. Therefore the compensation has to be high for the receiver so that he will take the proposed action a' over the most promising one a_{max}. On the other hand the compensation has to be low, as the proposer will see no benefit from making an offer. For this purpose we can change the mark up M_c to create a win-win situation for the agents.



(a) rewards with fixed walls **(b) rewards with random walls** **(c) agent 1 rewards with fixed walls**

Figure 4: Escape Room: Learning Process of cooperative and non-cooperative agents. (a) Agents with trading capability converge to the maximum, while non-cooperating agents fail and decrease to a minimum of -100. (b) The performance of agents with network compensations decreases as the learning process becomes more difficult. (c) Conditioning of agent 1 by agent 0, individual rewards of agent 1 decreases with Action Trading.

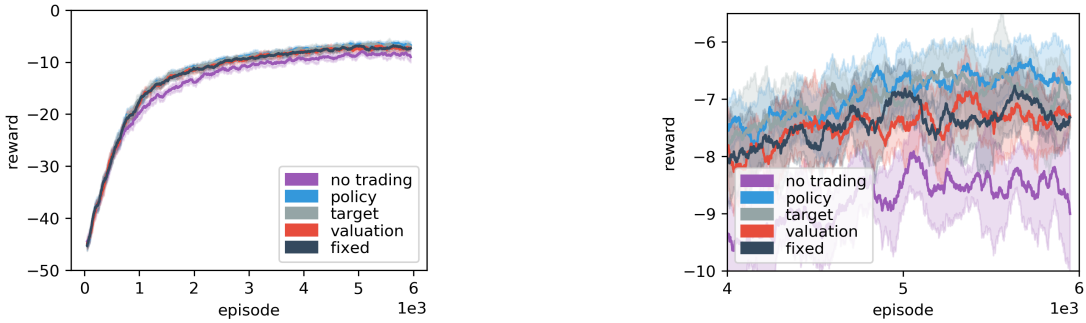
5 EXPERIMENTS

To evaluate the *Action Trading* approach we use the environments *Escape Room* and *Smart Factory*. As cooperation requires at least two entities, we use two agents in both domains. For every agent we use a *Deep Q-Network* (*DQN*) [5], with which they are able to learn to maximize their reward. These *DQN*'s consist of three hidden linear layers with *Exponential Linear Unit* (*ELU*) activation: 128, 64 and 32. Receiving a observation of the environment, the *DQN* outputs an action, which the agent performs. The *DQN*s use the *Adam algorithm* with a learning rate of $5e^{-4}$ as an optimizer. For the discount factor γ we use 0.95, because the action trading approach has to consider future steps. To achieve sufficient exploration, we train the agents over 6000 episodes, each consisting of a maximum of 100 time steps. Therefore we use the *decaying ϵ-greedy exploration* with $\epsilon_{max} = 1.0$ which decreases linearly with $\epsilon_{dec} = 6,67e^{-6}$ to a minimum of $\epsilon_{min} = 0.01$. We use 10 independent runs to evaluate the results of the 5x5 grid environments. In the *Smart Factory* the priorities of the agents are selected randomly each episode. Both agents have to process 3 random generated tasks and receive a reward of 1 for completing one. The high priority agent takes a 0.5 penalty every step, while the low priority agent only loses 0.02. In the *Escape Room* environment we keep the attitudes of the agents fixed over the complete run. The agent who wants to leave the room, agent 0, gets a penalty of 1 every time step he stays in the environment. But if he is able to reach the door, he gets a reward of 10 and the episodes ends. The other agent, agent 1, receives no penalty for staying in the room. But he gets punished with 1 for pulling the lever. Additionally, we investigate the performance of the agents when we fix the door direction or generate it randomly.

With regard to *action trading*, we want to examine the impact of various compensation amounts on the agent's performance. For this purpose we use different networks for the Q-values in 3: *Valuation Network*, a previously trained network based on non-cooperating agents. *Policy Network*, the currently online learning network. *Target Network*, the online network that is fixed and updated after a certain update period. In addition, we use a handmade *fixed* value to better evaluate the performance of the previous mentioned networks. For all network based compensations we use the mark up $M_c = 1.1$, as we not only want to make the proposed action indifferent to the most promising action but motivate the agent to follow the offer. In regards to the handmade fixed compensation we use $C_f = 2$ which exceeds any rewards an agent is possible to gain in the environment.

5.1 Escape Room

In the *Escape Room* agents have to exit a room through pulling a lever. However, only one agent can pull it and has to be motivated to do so. For the first experiment in the Escape Room we keep the direction of the wall fixed and observe that cooperative agents outperform non-cooperative ones. During the first 1500 episodes the reward of all agents increases due to the exploration phase, as the actions are chosen randomly with the probability of ϵ. As the ϵ continues to decrease over the course of the episodes, we can observe a constant decrease in the reward for non-cooperative agents to a minimum of -100. However, the reward for all cooperative agents increases. This can be explained with the design of the Escape Room, as the agent who pulls the lever is punished for doing so. In order to pull the lever the agent has to see a benefit from this action. The non-cooperative agents can not allocate reward to overcome this hurdle and the agent will therefore not pull the lever. As cooperative agents can use the action trading mechanism to give their own reward to the other agent, they can overcome the hurdle together and solve the escape room. However, there are differences in the rewards for the various compensation calculations. With the valuation networks the agents are doing worse compared to the other compensation approaches. As the valuation network is based on non-cooperative agents, the Q-values from the networks are not representing the correct expected rewards of the trading agents. The compensation amount is therefore not accurate and is harming the willingness of the agents to trade. This is also reflected in the graph 4a, as the curve alternates and has a rather large confidence interval. For the fixed compensation of 2 the performance is better, as the compensation amount exceeds the penalty for pulling the



(a) rewards of agents

(b) rewards of agents close-up

Figure 5: Smart Factory: Learning Process of cooperative and non-cooperative agents. (a) Cooperating agents converge faster to the maximum than non-cooperating ones. (b) Trading agents perform significantly better than non-trading agents.

lever 2. The agent can therefore be motivated to pull it by a trade, which results in a gain of $2 + -1 = 1$ reward. However, since the compensation is handmade, it is not optimal for both entity. The curve in 4a therefore converges more slowly to the maximum compared to the policy and target networks. Observing the rewards of the policy and target networks compensations, we examine a near optimum performance of the agents after 2000 episodes. This can be explained by the high willingness of both agents to trade. Moreover, the performance of the two networks are very similar, as the target network is a temporal fixed version of the policy network. Both the proposer and the receiver see a great advantage in trading, as the networks are directly linked to the current expected reward. As a result, the compensation for the lever pull with a mark up of $M_c = 1.1$ just exceeds the expected reward of the agent by 10 % 2. However, there is also a problem: Since the agent is motivated, he will soon perceive the pulling action as his most promising action. But as the compensation depends directly on the same network, or is delayed for the target network, the agent is now compensated by the proposer for the lever pull with 0. This does not cover the penalty and the agent gets $0 + -1 = -1$ reward for the pulling action, which ultimately puts the pulling agent in a worse situation than if he had not performed this action, see 4c. We can therefore assume that the policy and target compensation calculations do not lead to win-win situations, but to pure conditioning by the proposer. However, these two calculations are not the only one to suffer from conditioning, seen in 4c. This is caused by the proposer, as he soon realizes during the learning process that he no longer needs to make an offer in order for the receiver to still perform the non-existent proposed action.

Using randomly generated wall directions for the second experiment, we observe different behaviour among the cooperative agents 4b. While the non-cooperative agents still fail in solving the escape room, the performance of the various networks has decreased. For the fixed compensation the performance remains similar to the first experiment. However, the confidence intervals of the results are significantly wider for all cooperating agents. This can be explained by the learning process of the agents, as they have more difficulties in solving the environment and as such the performance of all network compensations has decreased. Especially the policy and target networks suffer from this, as the compensation depends directly on the online learning networks.

5.2 Smart Factory

The agents in the *Smart Factory* must complete tasks as fast as possible through processing machines. The environment is therefore not a cooperative but an optimization problem. We want to compare the non-trading with the trading agents to see if cooperation through the action trading mechanism helps to maximize the rewards of the agents. Using the same compensation calculations from the Escape Room domain we observe all trading agents outperform the selfish ones, seen in 5a and 5b. This is due to the agents trading behavior and the agent with the higher priority can therefore process all of his tasks faster. However, there are some differences in the performance of the various compensations: The agents with fixed and valuation networks for the calculation of the compensation perform worse than those with policy and target networks, see 5b. This can be explained as the fixed compensation is handmade much like in the Escape Room and the valuation networks are based on non-cooperating agents who produce sub-optimal values for the trading agents. The policy and target networks on the other hand gain the Q-values from the online learning networks and are therefore almost the same, since the target networks are a time-bound version of the policy networks. As a result, agents with the policy and target networks can more easily motivate each other to trade and thus perform better overall than the other compensations. Nevertheless the agents condition each other like in the Escape Room 4c.

6 CONCLUSION

In conclusion, the concept of trading agents can also be applied to the *Smart Factory* and *Escape Room* environments. Here, the cooperative agents achieve significantly more reward than the non-cooperative ones with various compensation calculations. In the *Smart Factory*, all trading agents perform well 5a: In this domain the valuation networks can be used effectively, because the pre-trained non-cooperative agents already deliver appropriate Q-Values. Also the hand-selected fixed compensation of $C_f = 2$ shows clear success in this domain, as the value exceeds the expected reward of the

receiver 2. Finally, the policy and target networks are very similar, as the target network is a temporary fixed variant of the policy network. Agents achieve the highest rewards with these two network compensations. One reason for this is that the proposer conditions the receiver to perform actions for him without any compensation. This can also be seen in the *Escape Room* environment without random wall direction 4a: Although the agents with policy and target network outperform all other agents, it is clearly visible in 4c that the proposer conditions the receiver. The result is not a win-win situation, since the curve of agents with networks is much lower than that of the non-cooperative agents. Also, compensation with the valuation network works relatively poorly in this environment compared to other networks. This is because the compensation depends on the pre-trained non-cooperative agents, but it does not match the expectations of cooperative agents. For the fixed compensation it can be observed that the curve performs well in both the non-random and random *Escape Room* and rises steadily, although more slowly, to the maximum. In the *Escape Room* with random walls it can be observed that the compensation calculation by the networks also depends strongly on the success of the online learning agents. Here, 4b, all curves based on networks show worse performance than the fixed compensation or in the case that the wall is not randomly generated. Summing up, it can be said that cooperative agents also perform better than non-cooperative agents. The handmade compensation is solid in any environment, but as a drawback has to be picked by hand. For this purpose, different approaches for improvement were analysed: Valuation networks perform well in domains where normal agents already perform reasonably good, but fail in other environments where non-cooperative agents perform poorly. In this case, other compensation calculations can help out: For example, agents with policy and target networks can achieve great results depending on the learning success, but the receiver suffers significantly from conditioning of the proposer. For the action trading it can be said that the different compensations in this work are not universally applicable to different environments and that each method to calculate the compensation has its disadvantages.

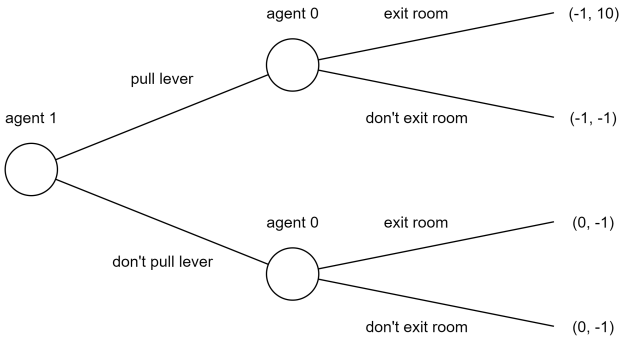


Figure 6: Decision making of the agents in the escape room: Agent 1 expects a higher reward from the action *don't pull lever* than from *pull lever*, which causes agent 0 to suffer from the decision of agent 1.

REFERENCES

[1] Yoram Bachrach, Richard Everett, Edward Hughes, Angeliki Lazaridou, Joel Z. Leibo, Marc Lanctot, Mike Johanson, Wojtek Czarnecki, and Thore Graepel. 2018. Negotiating Team Formation Using Deep Reinforcement Learning.

[2] Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge. 2011. Computational Aspects of Cooperative Game Theory. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 5. https://doi.org/10.1007/978-3-642-22000-5_1

[3] Sarit Kraus. 1993. Agents Contracting Tasks in Non-Collaborative Environments. In *AAAI*.

[4] Adam Lerer and Alexander Peysakhovich. 2017. Maintaining cooperation in complex social dilemmas using deep reinforcement learning. *CoRR* abs/1707.01068 (2017). arXiv:1707.01068 http://arxiv.org/abs/1707.01068

[5] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013). https://arxiv.org/pdf/1312.5602.pdf

[6] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei Rusu, Joel Veness, Marc Bellemare, Alex Graves, Martin Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518 (02 2015), 529–33. https://doi.org/10.1038/nature14236

[7] Kyrill Schmid, Lenz Belzner, Thomas Gabor, and Thomy Phan. 2018. Action Markets in Deep Multi-Agent Reinforcement Learning. In *Artificial Neural Networks and Machine Learning (ICANN 2018)*, Věra Kůrková, Yannis Manolopoulos, Barbara Hammer, Lazaros Iliadis, and Ilias Maglogiannis (Eds.). Springer International Publishing, Cham, 240–249.

[8] R.S. Sutton and A.G. Barto. 1998. *Reinforcement learning: an introduction*. MIT Press, Cambridge, MA.

[9] Jiachen Yang, Ang Li, Mehrdad Farajtabar, Peter Sunehag, Edward Hughes, and Hongyuan Zha. 2020. Learning to Incentivize Other Learning Agents. arXiv:2006.06051 [cs.LG]