CITS3001 Report: Bayesian Opponent Modelling and the Effect of *Ex Ante* Collusion Amongst Spies

# Introduction

## Background

Resistance is a hidden role game of 5-10 players in which each player is designated a role as either a resistance or a spy. The goal of resistance is to ensure missions go ahead and succeed. The goal of spies is to either prevent missions from going ahead by voting against them or by ensuring that enough spies are on the mission to sabotage it causing it to fail.

Resistance players are initially unaware of the spies within the group however spies are provided with information as to who the other spies are and therefore know the type, either resistance or spy, of all players in the game. As moves are made simultaneously the resistance and spies are both playing a game of imperfect information. It is however the case that spies have more information at the start of play than the resistance. The resistance members must try to deduce the players that are spies and prevent them from being included in missions to prevent sabotage.

## Research Question

The goal of this project was initially to create an agent using Bayesian reasoning to build an agent that could outperform a deterministic agent both as a member of the resistance and as a spy. All forms of the game were to be assessed using each game type from five to ten players. Using the win/loss as a series of Bernoulli trials to determine whether this goal had been achieved.

Various scenarios were created to test the effect of the changes. First was to bench mark the win/loss for the agent acting as both spy and resistance. Secondly the various agents would be assigned a role based on their type e.g. all spies are Bayesian agents and all resistance deterministic and assess the changes based on specific roles. The last part was to check how a single agent acting as either role could affect the outcome of a game consisting of entirely different agents.

In the process of building the deterministic agent it was found that collusive behaviour could be built into the spies standard operating procedure allowing them to agree on which spies on a mission would sabotage it without requiring communication. This would have the benefit of minimising the footprint of sabotage on a mission by ensuring only the minimum number of spies required would betray the mission in effect minimising the suspicion for all agents on the mission by maximising the uncertainty over betrayal.

At this point the project question was split between building an agent using Bayesian reasoning and determining how collusion would affect an agent designed to beat that agent with and without collusion.

# Literature Review

## Agent Selection

Methods for playing The Resistance using hard coded deductive logic provide a solid foundation for implementing a base set of rules that are fail safe and should exist in conjunction with other methods. An example of this logical play is to decide that in the event that the number of agents on the mission is equal to the number of betrayals all of the agents on the mission are spies. This type of logic can be determined easily through playing the game with pre-existing go-bies for this type of game play, grouped as ‘Expert Rules’ can be found extensively in Taylor (2014). However, the majority of information in The Resistance does not provide definitive evidence preventing it from being a complete method for playing the game.

Various search methods covered in the unit so far are feasible for use in creating an agent to play The Resistance as there is a finite number of possible moves and rounds. Methods such as A\* and Minimax are possible options for use in a game like The Resistance however there is a large state space for even the five-player game estimated by Serrino *et al* (2019) as 10^56. As the game type moves towards ten players the search space becomes significantly larger as the combinations of teams, votes and outcomes for each mission increase. Given that this project looks at the five to ten player state space it would be difficult to create efficient search. There does not appear to be a simple heuristic for reducing the state space in The Resistance as we are rarely certain as to what missions or vote outcomes actually mean until much later in the game. For this reason, it was decided that other methods would prove easier to implement with better results.

Neural networks seem to be an obvious choice for AI agents given their popularity, the access to libraries and with recent research showing that they work well specifically for The Resistance (Serrino *et al*2019). However, there is not much we do not know about the game play for The Resistance running information through a neural network seemed like overkill. Further to this, access to compute resources and the learning curve in setting it all up given the time constraints was deemed prohibitive. Monte Carlo Tree Search seems to be the one of the most popular methods for playing hidden role and hidden information games (Reinhardt 2020; Taylor 2014; Whitehouse 2014). Given the amount of time required for this project setting this up was also deemed infeasible.

Bayesian reasoning appeared a

The priors given were chosen randomly and configured until they worked however there is undoubtedly a set of values that will maximise the success of the agents. It was initially thought that this could be optimised by using a simple hill climbing search or possibly with genetic algorithms. Unfortunately time was not on the side of this plan so it was left off.

## Collusion

Research exists within the realm of generating collusion within models. However the use of collusive tactics within simultaneous games is an approach that does not appear to have much literature as it side steps the idea of fairness within the game. This seems to put it outside the bounds of most research which focuses on maximising play against a fair opponent.

Collusion is a common strategy in games of strategy particularly for gamblers looking to make money from a casino. In popular culture there are numerous books, films and examples of a team of players using collusive tactics to beat the house, the most famous of which is likely the MIT Black Jack Team. Research has also shown that this is an effective model for winning (Ullman 2015; Zehnder 2012). There is however a significant body of work in regards to detecting collusion in games as it is relevant to both online games and gambling (Mazrooei *et al* 2013, Yan 2010).

There is some research into collusion using imperfect information games in AI however this focuses on having imperfect information at each stage (Farina *et al* 2018). For example working towards a common goal of wins while not knowing what hands the colluding players holding poker. This differs quite significantly to the example in The Resistance where each player has perfect information as to what other players are spies and whether they are on the mission and can therefore decide before hand who will sabotage a mission should all spies be on the mission using, for example, player number as rank that will determine the saboteur.

Similarities do exist in the literature on the generation of predator prey models in which the predators/prey are able to cooperate to hunt/avoid prey however this exists more within the bounds of multi-agent learning rather than preconfigured cooperation (Stone and Veloso 2000).

# Selected and Design of Methodology

The Resistance is a game of determining roles given uncertainty in actions. Probabilistic methods are appropriate as they provide us insight into the likelihoods of events given what we have observed. As resistance members are only ever in a position to understand events through observation rather than through perfect information.

Bayesian inference was selected as a simple to understand method that would allow for an understanding of state immediately before and after events in the game. This meant the effect of changing any hyperparameters such as the probability that an agent would betray a mission given no other information could be adjusted.

In understanding the probability that each agent is a spy, we can then determine what the probability of a mission succeeding or failing is given the composition of the mission. This allows us to determine which missions we should propose as a resistance member as we are trying to maximise the likelihood of the mission succeeding. Likewise, it allows us to vast votes based on our assessment of whether the mission will succeed or fail. Further to this, if we notice that an agent that we consider more trustworthy assigns agents we consider less trustworthy, we now have an option to update how much we trust them based on this new information.

As a spy the agent can use this same technique to their advantage as they need to understand which missions will appear most acceptable to resistance agents. It is advantageous for spies to know which missions will ‘appear’ to succeed whilst having perfect information about the composition of spies on the mission. With this information the spy is able to put forward a mission that initially appears like it will succeed but will likely be betrayed. In knowing this it is also possible for spies that know they are untrustworthy to make assessments of resistance that appear untrustworthy and suggest them for missions in order to bluff other agents assessment of them.

## Deterministic Agent

The first part of the design involved implementing a purely deterministic agent that would act according to pre-configured plans when it was certain, when it was not it would act using hard coded probabilities. The pre-configured settings involved determinations for all types of play both resistance and spy.

#### Mission Outcomes

When a resistance member was on a mission and all other members of the mission betrayed the mission they would confirm there status as spies and would no longer vote for any missions they proposed nor for any missions they were included on except the final round of voting. In the event that all players on a mission betrayed the mission they would considered burnt. Any player proposing burnt players would also become burnt as they would be considered untrustworthy to user these players.

#### Votes

The deterministic agent would vote against known spies whether they were on the mission or a proposer. When they had no evidence not to they would vote true. As a spy, the agent would always vote for missions with other spies unless that spy was burnt. It also voted against missions that included resistance agents that knew their status as a spy. Resistance members were chosen randomly for missions.

#### Propose Mission

As resistance the Deterministic agent would propose a mission with players they could not confirm were spies. As a spy the agent would always include itself and when required another random spy that was not burnt if the mission needed more than one saboteur to fail.

#### Betrayal (Standard)

When either a betrayal would result in a win or on the final round the spy would always sabotage the mission. When there were not enough spies to sabotage the mission the agent would not act.

#### Betrayal (Non-Collusive)

When not acting under collusion and the standard rules were not met the player would use the tactics here.

In rounds 4 and 5 when the only spy on the mission the player would always betray the mission. In earlier rounds the spy would use a low probability random choice to determine whether they would betray the mission

#### Betrayal (Collusive)

When acting with collusion the agent would instead get the list of spies on the mission. If the agent had been assigned the maximum player number of the spies on the mission the agent would betray the mission. In the event that they had any other number they would not. This guaranteed that the mission would fail but the footprint would be as small as possible. With collusion on it became effectively impossible for the resistance to win a game unless they went through three games without a spy on the mission by chance.

## Probabilistic Agent

The Probabilistic Agent is an extension of the Deterministic Agent in which

[We are using a model of another agent to predict it’s goals and beliefs.](https://www.cs.utexas.edu/~larg/ijcai17_tutorial/multiagent_learning.pdf)

The side goal of this is to be suitable for ad hoc team work with other agents during the competition round

“Game Strategies and Decision Making” (Harrington)

Test against the baseline. Test against random to make sure it is not modelled specifically for the baseline. Test the baseline against random to compare

## Rationale of Selected Technique

## 

##NOTE##

It is thought that significantly more time could be put into the assessment of spies and the modelling around risk. This said, the gains made by the resistance could likely then be factored into further collusion by spies.

## Validation of Agent Performance

Bernoulli Trials and Hypothesis Testing

## Bibliography

[Cowling, P.I, Powley, E.J, Whitehouse D, (2015) ‘Emergent Bluffing in Inference with Monte Carlo Tree Search’, IEEE Transactions on Computational Intelligence and AI in Games, 4(2) pp.120–143](https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7317927)

Dutta, P.K. (1999). Strategies and Games: Theory and Practice. Cambridge, MA, The MIT Press. Chapters 14-15 for Repeated Games. Chapter 20 for Games with Incomplete Information.

[Farina, G, Celli, A, Gatti, N, Sandholm, T, (2018) ‘Ex ante coordination and collusion in zero-sum multiplayer extensive form games’ Advances in Neural Information Processing Systems, Vol 31](https://proceedings.neurips.cc/paper/2018/file/c17028c9b6e0c5deaad29665d582284a-Paper.pdf)

[Mazrooei, P, Archibald, C, Bowling, M, ‘Automating Collusion Detection in Sequential Games’ Proceedings of the 27th AAAI Conference on Artificial Intelligence](https://www.aaai.org/ocs/index.php/AAAI/AAAI13/paper/download/6448/7235)

[Reinhardt, J. (2020) ‘Competing in a Complex Hidden Role Game with Information Set Monte Carlo Tree Search’ arXiv:2005.07156v1 [cs.AI]](https://arxiv.org/pdf/2005.07156.pdf)

[Serrino, J, Kleiman-Weiner, M, Parkes, D.C, Tenenbaum, J.B,  *et al* (2019) ‘Finding Friend and Foe in Multi-Agent Games’ arXiv:1906.02330v1 [cs.LG]](https://arxiv.org/pdf/1906.02330.pdf)

[Stone, P. and Veloso, M (2000) ‘Multiagent Systems: A Survey from a Machine Learning Perspective’, Autonomous Robots, Vol 8, pp.345-383](https://link.springer.com/content/pdf/10.1023/A%3A1008942012299.pdf)

[Szita, I, Chaslot, G, Spronck, P, (2009) ‘Monte Carlo Tree Search in Settlers of Catan’, Advances in Computer Games pp.21-32](https://link.springer.com/content/pdf/10.1007%2F978-3-642-12993-3.pdf)

[Taylor, Daniel P. (2014) ‘INVESTIGATING APPROACHES TO AI FOR TRUST-BASED, MULTI-AGENT BOARD GAMES WITH IMPERFECT INFORMATION WITH DON ESKRIDGE’S “THE RESISTANCE”’](https://teaching.csse.uwa.edu.au/units/CITS3001/Resistance.pdf)

[Ullman, R.J, (2015) ‘Optimal Poker Strategies Under Collusion’ PhD Thesis, Princeton University](http://bobbyullman.com/files/poker/thesis-fall15.pdf)

[Whitehouse, Daniel (2014) ‘Monte Carlo Tree Search for games with Hidden Information and Uncertainty’, PhD thesis, University of York](https://core.ac.uk/download/pdf/30267707.pdf)

[Yan, J. (2010) ‘Collusion Detection in Online Bridge’ Proceedings of the 24th AAAI Conference on Artificial Intelligence](https://www.cs.huji.ac.il/~jeff/aaai10/02/AAAI10-265.pdf)

Young, H. Peyton (2007) ‘The Possible and Impossible in Multi-Agent Learning’, Artifical Intelligence (AIJ), Vol 171, pp. 429-433

[Zehnder, B. (2012) ‘Collusion in Online Poker Pays Off’ Bachelor’s Thesis’, Swiss Federal Institute of Technology](https://pub.tik.ee.ethz.ch/students/2012-FS/BA-2012-03.pdf)