Mechanism Design in *The Resistance*

Mike Ambrose, Barak Gila, Rudy Laprade
University of California, Berkeley

Abstract

We look at The Resistance*, a Mafia-like hidden information game, under the lens of mechanism design. By modeling the choice of mechanism as a subgame with a restricted strategy set, we create a solution which provides a strategy for any state of the original game. Furthermore, we prove that cooperation between the "good" players (Resistance) in the form of this self-imposed mechanism is enforceable without the need of an overseer and maximizes Resistance chance of victory. To calculate the optimal mechanism and thus the optimal strategy, we implement the method of counterfactual regret minimization (CFR) to give an ϵ-approximate Nash equilibrium for the full game.*

I. Introduction

Games involving hidden information are popular for both competitive and casual play. In hidden information games, the behavior of the other players becomes as important to gameplay as the current state of the game. Interesting behaviors like bluffing and lying emerge, along with the necessity of "reading" an opponent to determine if they are bluffing. Unsurprisingly, imperfect information games are typically far more difficult to analyze than perfect information games. This is due to the sheer number of states and corresponding strategies. To solve reasonably simple two-person games with hidden information is cutting-edge. A recent result in this field was the solving of Heads-up Limit Hold'em Poker, a limited version of the popular poker games Texas Hold'em[2]. Notably, this result is only for the two-player version of the game, as the complexity quickly spirals out of control as more players are added. Solving any larger game usually involves taking advantage of symmetry and creating simplified "approximate" versions of the game [3].

The Resistance is a Mafia-like party game which is very popular among board game enthusiasts, with over 100,000 copies sold [1]. Mafia is a well-studied game with strong results concerning the optimal size of the Mafia[6]. However, because players are not eliminated on each turn, the recursive analysis done on Mafia does not work on *The Resistance*. Furthermore, *The Resistance* is an enormous game. The size of a game is often measured in number of distinct information sets. An information set is the set of all information which one player knows. For instance, in a game of Texas Hold'em, a player's information set would include the cards in their hand, the cards on the table, and every betting action which any player had taken. The largest game solved to date has around 10^{13} distinct information sets[2] and took over 900 core-years of computation. *The Resistance* has 7×10^{63} information sets[1]. Therefore, even the new methods utilized for solving Texas Hold'em problems are nowhere near sufficient for *The Resistance*. We instead look to exploit the rules of the game to propose a mechanism which all players are motivated to follow.

In Section II, we describe the mechanics of *The Resistance* for those unfamiliar with the game. In Section III, we describe how to view the game as a mechanism and how to find the best mechanism for the game. Section IV describes why the mechanism is optimal for the Resistance and is thus self-imposing. Section V describes the practical result of our paper, the use of linear programming and the method of counterfactual regrets used to find a $\epsilon-$approximate Nash equilibrium to the mechanism. Section VI discusses the results of these computational methods as well as how we've visualized and published these results. Finally, Section VII discusses the specifics of the mechanism computed as well as future work on imperfect games.

II. Resistance Mechanics

In order to understand the mechanism, it is important to understand the basic rules of *The Resistance*. *The Reistance* can be played with 5 or more players, but for the purposes of this paper we will consider only the 5 player version. The game begins with 2 players being randomly chosen to be

[1] Before symmetries are taken into account. We will discuss symmetries more later - they let us cut the number of information sets substantially. However, their utility is fairly limited for the full game, and after symmetries the number of distinct information sets would still be nowhere near feasible.

Spies. Imperfect information arises here as each Spy knows the identity of the other Spy, but the 3 other players (the Resistance) are clueless to the roles of anyone. The Spies and the Resistance form the two teams for the game.

The Resistance is a five-round game. On each round, the current player proposes a "mission", which is a fixed size [2] subset of the players. Every player then publicly votes on whether to accept or reject the mission. If the majority of players vote to reject the mission, the next player proposes a mission [3]. If the majority of players vote to accept the mission, then the mission is executed: every player on the mission privately votes either PASS or FAIL. The mission passes (the Resistance wins the round) if the votes are unanimous PASS, and fails (the Spies win the round) if there is at least one FAIL vote. Note that all players do see the result of each mission (how many passes/fails), but votes are anonymous. At this point, the next round begins.

The game is decided on the result of the missions. If the game reaches 3 passed missions, the Resistance wins. If the game reaches 3 failed missions, the Spies win. Therefore, the Spies have a desire to fail missions when they can (since that is their only path to victory), but occasionally choosing to pass a mission can confuse the Resistance about the true locations of the Spies. The members on the Resistance, on the other hand, always pass the missions.

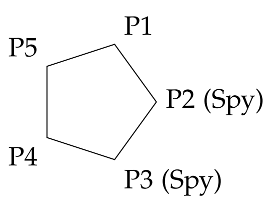

Figure 1: *Example Resistance layout*

We will quickly go through an example round. Imagine that the game has just begun. P1 is the first person to propose a mission, and he proposes the mission of {P1, P5}. Now every person at the table publicly votes on whether to accept or reject the mission. Imagine that the votes are $\{A, R, R, R, A\}$ for players 1 through 5, respectively. Since the majority voted to reject, the next player on the table, P2, proposes a mission. P2 proposes the mission of {P1, P2} and the votes are $\{A, A, A, R, R\}$. Since the mission was accepted with a majority, each of P1 and P2 (the two players on the mission) privately decide whether to pass or fail the mission. P1 is Resistance and thus will always pass; however, P2 is a Spy and decides to fail the mission. All players then learn that the mission has failed, and P3 begins the next round by proposing a mission.

As noted earlier, all players can see the number of PASS and FAIL votes on a mission. If both spies ever voted to fail a mission, the resistance members could determine that anyone not on the mission must be in the Resistance, which would be disastrous for the spies. In our analysis, we assume that this never happens: if both spies are on the mission and they intend to fail it, they will ensure that only one of them votes FAIL [4]. Since for the spies, playing FAIL votes is dominated by playing a single FAIL vote, this assumption does not affect the strategies we compute.

III. Mechanism Design

1. Proposed Mechanism

The Resistance lends itself well to a collaborative solution. Since there are more Resistance members than there are Spies, any mechanism can be self-enforcing if it is beneficial for the Resistance. We will show this rigorously later; however, it is enough to note that, even if all Spies deviate from the mechanism, they do not impact the chosen mission.

We propose the following framework for a mechanism: each round, a publicly visible and tamper-proof random number generator is used to select a mission M from a mixed strategy $\mathcal{S}_Q$ based on our current public state Q. Everyone votes to accept the mission and so Spies choose to either pass or fail the mission (if there is a Spy on the mission - otherwise, the mission passes by default). Based on the outcome of M, the next round the mission would be selected from a new mixed strategy corresponding to the new public state. Each of these mixed strategies could be completely distinct and must be computed separately.

2. Mechanism as Game

This mechanism creates a new 2-player game out of the original game. The players of this game are the Resistance

[2]Depending on the mission. On the first round, 2 players are in the mission. On the second round, 3 players are in the mission. On the third round, 2 players are in the mission. On the fourth and fifth rounds, 3 players are in the mission.

[3]For each round, if there are four rejections, the fifth proposed mission is accepted without voting.

[4]One way to accomplish this is for players to pre-arrange a priority order for voting to fail; for example, whomever's name comes first alphabetically.

and the Spies. Each strategy for each side represents a complete plan for every possible state of the game. For instance, here is an illustration of a few levels of a Resistance strategy

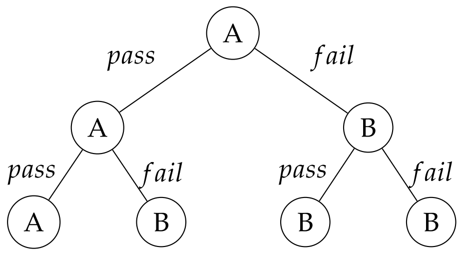

Figure 2: *Resistance pure strategy*

This corresponds to a strategy where the only possible missions to propose are A and B, where the first proposition is always mission A. This snippet of a strategy is for only two rounds. The actual decisions here are the mission (whether to propose A or B at any given node). For instance, if the Spies response to mission A was to fail, this pure strategy would propose mission B. A spy strategy would look similar, but must take into account the location of the spies.

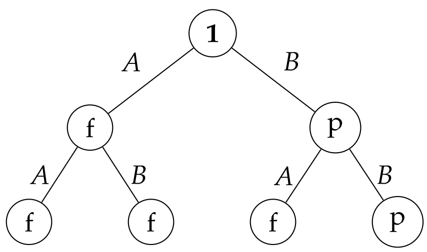

Figure 3: *Spy pure strategy*

where here, **1** corresponds to a specific location of Spies. Any Spy pure strategy has a best response for any configuration of Spies. We will discuss symmetries when we solve the mechanism, but for now, this is a simple representation which completely captures the possible behaviors of both Spies and Resistance. Evaluating the result of two pure strategies playing each other can be done easily by following the tree for each initial configuration of spies and seeing how many of the resulting games are won by the Resistance. For utilities, we represent this game as a zero-sum game, where the utility of the Resistance is the probability that they win and the utility of the Spies is the negative of that probability.

This is not quite sufficient, since the Resistance cannot just play the full tree immediately, or else the Spies would no longer have to play a mixed strategy and could instead just choose their one best response (this is akin to going first in rock-paper-scissors). In the Appendix, we show how to choose S_Q from a mixed strategy of trees. For now, it suffices to say that we can achieve the value of this game in practice.

IV. Mechanism Dominance

While our mechanism does correctly implement the Nash equilibrium, it is only meaningful if players are motivated to play the mechanism. Unlike true mechanism design, in which a designer imposes costs on different outcomes, our mechanism is unsupervised and must be individually desirable by each player.

First, we will demonstrate that, if the Resistance players obey the mechanism, the Spies have no incentive to deviate - in other words, following the plan strictly dominates deviating from the plan. To prove this, we use the game property that missions pass if and only if they have majority support. Even if all Spies simultaneously deviate, they do not impact which missions pass. If a Spy proposes a mission that does not correspond to the mixed strategy, all Resistance players will vote to reject it, so it will not pass. Therefore, since Spies can make no difference in the actual missions proposed, the only result of deviating from the mechanism is differentiate themselves from the Resistance players who obey the mechanism (and if the spies reveal their identities in this manner, they will surely lose the game; at that point the resistance players could avoid putting them on missions).

However, this relies on Resistance players obeying the mechanism also. In order for Resistance players to obey the mechanism, they must believe that their probability of success while playing with the mechanism is higher than their probability of success without the mechanism. This is not trivial and does not derive directly from the definition of the mechanism. If there existed some non-distributed strategy, where each player formed their own opinions and proposed missions independently, with a higher chance of Resistance success than the mechanism, the Resistance players would have no incentive to follow the mechanism. This proof of dominance is the main theoretical result of the paper.

Before we jump into the proof, we give a brief summary. The proof considers the value of the game when all

Spies publicly behave as Resistance. We demonstrate that, in this case, the only relevant parts of the game are the missions which pass and the results of those missions. We then demonstrate that proposing missions according to our mechanism maximizes the likelihood of winning. Since the Spies could choose to behave as Resistance, the game must be at least as good for the Spies as the mechanism value; therefore, the Resistance maximize their chance of winning by playing the mechanism.

1. Playing a Subset of Spy Strategies

The rest of the proof will revolve around Spies playing a specific subset of their strategies. Specifically, we want the Spies to have "temporary amnesia" such that they believe they are Resistance when they propose or vote (the only two public, non-anonymous actions). This is feasible since it involves playing some subset of possible Spy strategies.

We define the value of the game $V(G_{\mathcal{S},\mathcal{R}})$ (with $\mathcal{S},\mathcal{R}$ strategy sets for Spies and Resistance) as the equilibrium with the highest Resistance win rate.

Lemma 1: The value of the full game is at most the value of the game when Spies publicly imitate Resistance.

Proof: When the Spies imitate Resistance, they play some strategy subset $\mathcal{S}' \subseteq \mathcal{S}$. We wish to demonstrate that $V(G_{\mathcal{S},\mathcal{R}}) \leq V(G_{\mathcal{S}',\mathcal{R}})$. Assume for contradiction that $V(G_{\mathcal{S},\mathcal{R}}) > V(G_{\mathcal{S}',\mathcal{R}})$. In the game with $\mathcal{S}'$, the Spies play some mixed strategy $\mathcal{M}_{\mathcal{S}'}$ over $\mathcal{S}'$ which gives them $V(G_{\mathcal{S}',\mathcal{R}})$. Since $\mathcal{S} \supseteq \mathcal{S}'$, the Spies can also play $\mathcal{M}_{\mathcal{S}'}$ in the full game, and the Resistance's best response has value $V(G_{\mathcal{S}',\mathcal{R}})$. This is a better value for the Spies, so they can profitably deviate, which is a contradiction.

2. Information and Imitation

Intuitively, our goal is to show that Spies have strictly less power when we play the mechanism. To formalize this, we consider the game where Spies publicly act as Resistance. We can represent the state of a game as some prefix of

$$\{P_{11}V_{11}P_{12}V_{12}\cdots P_{15}R_1, P_{21}V_{21}P_{22}V_{22}\cdots\}$$

where P_{ij} represents the jth mission proposed on round i, V_{ij} represents the votes of every player on the jth mission proposed on round i, and R_i represents the result of the passed mission on round i. This information set assumes that every round goes through the full 5 proposals, which is certainly not necessarily true - however, it is trivial to insert "null" missions for the remaining proposals which represent no proposal due to a passed mission. For future proofs, we will refer to the mission which actually went through (either passed by votes or was last proposed) as M_i. We extend our previous definition of game value to describe *states* instead of *strategies* - that is $V(\{P_{11}V_{11}\cdots\})$ is the maximum equilibrium Resistance win rate when $\{P_{11}V_{11}\cdots\}$ has been observed.

Because the Spies play exactly the same as Resistance, we have the following useful lemma:

Lemma 2: The value of any state S depends only on the passing missions M_i and the results V_i

Proof: We will proceed with another proof by contradiction. Assume there are two states S and S', where both S and S' have the same values for all M_i, V_i but have some other difference. Furthermore, without loss of generality, let $V(S) > V(S')$.

In order to have a different value, either the distribution of Spy locations must be different or the distribution of missions must be different. This is true because the value of a game is directly determined by mission proposals and whether or not the Spies fail them, and since the Spies are playing some fixed equilibrium strategy, whether or not Spies fail them is just a function of Spy locations. However, we prove that the distribution of Spy locations trivially cannot be different in S and S'. Since the votes and proposals are (by construction of the Spy strategy) independent of Spy location, changing a vote or proposal cannot change the distribution of Spy location.

Continuing with our proof by contradiction, this means that the missions accepted after S must be different than after S'. Without loss of generality, we say there exists some earliest round i such that the mixed strategy for state S of mission M_i, $\mathcal{M}^S_{M_i}$, must be different than its counterpart in S', $\mathcal{M}^{S'}_{M_i}$. Since $V(S) > V(S')$, consider the impact of playing $\mathcal{M}^S_{M_i}$ for mission M_i in S'. This is a valid mission proposal, and so for S to have higher value than S', this mission proposal must be less effective in S'. However, as described in the previous paragraph, this is only possible if the distribution of Spies is different, which is impossible.

Note that this lemma only holds when Spies play to imitate the Resistance (since otherwise the votes could potentially give information about the distribution of Spy locations). As a sanity check, the argument clearly does not hold to remove the mission proposals or results from the set of important information, since both of those give information about the location of spies.

3. Mission-Result Simplification

Now we consider a simplified view of the game in which only the missions and the results of the missions matter. We wish to prove that the series of missions proposed by the mechanism maximizes the value of the game. This is basically the definition of the mechanism, but we love lemmas, so we will give a proof anyways.

Lemma 3: If only missions and mission results matter, playing according to the distribution of the mechanism maximizes Resistance win chance.

Proof: This derives from a trivial proof by contradiction. Consider if there is some distribution at some state $\mathcal{M}'^{S}_{M_i}$ which leads to a higher value of the game. In that case, the mechanism would not be a Nash equilibrium to the reduced 2-person game, since the Resistance could profitably use that strategy. That is a contradiction to the definition of the mechanism, so the mechanism strategy must be optimal.

4. Final Combination

Finally, we will combine our abundance of lemmas to prove our main result.

Theorem: Playing the mechanism maximizes the value of the game.

Proof: We will prove that, when the Spies play to imitate the Resistance, the game has value equal to the mechanism value. By **Lemma 1**, that means that the whole game has value at most the mechanism value, meaning the Resistance should play the mechanism to maximize value.

When the Spies imitate the Resistance, by **Lemma 2**, the only parts of that state that matter for the value of the game are missions and results. By playing according to the mechanism, the Resistance maximize their win rate by **Lemma 3**.

This theorem does not rule out the possibility that other, non-mechanism strategies also achieve the mechanism value. All it says is that playing the mechanism will maximize win rate for the Resistance. However, if the Resistance are rational, all will agree that playing a strategy which is known to lead to maximal win rate is beneficial. Therefore, the mechanism is self-enforcing and a full description of the optimal value of the game.

V. Solving The Mechanism

1. Normal-Form Game

The obvious first approach to solving the game is to represent it as a normal-form game, with columns as Spy strategies and rows as Resistance strategies. Since our game is zero-sum, we can solve it using linear programming. However, as we've described previously, the number of strategies is exponential in the size of the information set. Naïvely, there are 10^{31} Resistance strategies and around 10^{150} Spy strategies. However, this number overcounts the unique strategies substantially, since there are many strategies which are symmetric.

Symmetries: The first (and easiest) symmetry to see is in the first mission. Since Spies are allocated randomly, it does not matter what the first mission is. We arbitrarily choose two positions to be the two players we send on the first mission. This helps us cut down on the number of information sets by an order of magnitude, which will be useful for CFR later. We can also use symmetries in other places throughout the game also. For instance, when the Spies decide whether or not to fail the first mission, their decision depends only on the number of Spies on the mission, not on their specific locations. This allows us to cut a substantial number of strategies; however, in the later rounds, often each player has been on a unique combination of missions, so symmetries cannot be used. Therefore, we need another method to cut down on the number of games.

Choiceless nodes: Throughout the tree, there are some nodes where one strategy strictly dominates another. For instance, if the Spies have already observed two fails and have the option to fail a mission, they always should. Failing the mission would lead to an instant victory, whereas passing the mission has some chance of leading to failure in the future. Similarly, the Resistance should never propose a mission which includes the entirety of a previous failing mission, since that mission is guaranteed to still have a Spy on it. Most importantly, on the last mission, the Spies always fail given a choice. This allows us to remove the majority of Spy decision nodes.

Early termination: The above numbers for Resistance and Spy strategies include numbers all the way down to the 5th round regardless of history. However, in the Resistance tree, the subtree rooted at the node after three passing or three failing missions is irrelevant, since at the point the game has already ended. Therefore, we can prune a substantial amount of the tree by ignoring any nodes after a decided point.

This allows us to cut the size of both strategy sets substantially, especially the Spies. However, even after symmetries, we find 10^{10} Resistance strategies and 10^3 Spy strategies, a very substantial reduction but still many orders of magnitude above what an LP solver can handle. Therefore, we can only solve simplified versions of *The Resistance* as a normal-form game. We strongly solve the three-round variant with LP, which we discuss in **Results**. However, we will have to use a different approach if we want to have a chance of solving the five-round game.

2. Counterfactual Regrets

The method of minimizing counterfactual regret (commonly called just CFR) is an algorithm for finding an $\epsilon-$approximate Nash equilibrium in imperfect information games. CFR attempts to minimize regret, since if both players play strategies with regret less than ϵ, the strategies are a $2\epsilon-$Nash equilibrium[4]. Instead of attempting to directly minimize regret, CFR minimizes a related quantity, counterfactual regret. The sum of counterfactual regret over time is an upper bound for regret, and minimizing it also minimizes regret and eventually leads to the approximate equilibrium. We will not completely re-describe the details of the method, which has been widely analyzed and used for solving poker games.

Unlike expanding to normal form, which was exponential in the number of information sets, CFR is polynomial in the number of information sets. Furthermore, our mechanism has very few information sets. After taking advantage of the above simplifications, the number of information sets is around 10^6, most of which are terminal (which do not require keeping state). With advanced hardware and optimized software, CFR can handle problems with 10^{10} hidden states. Therefore, our problem is a great candidate for CFR.

Since there are no publicly available implementations of CFR which are not poker-oriented, we implemented CFR ourselves. Our implementation is not *The Resistance*-specific – by changing certain functions to represent any imperfect-information game, our CFR will compute a solution to any game, within computational limits.

The two major problems with CFR are memory use and precision. Thankfully, since our problem is small, memory use was not a huge concern. By implementing the optimizations described by Neller and Lanctot[5] to avoid storing the complete history of strategies, our CFR uses the same amount of memory regardless of the number of iterations. Even in our small problem, using default Python floats very quickly lead to severely inaccurate results, due to accumulated numerical rounding errors. Instead, we used Python's Decimal class, which allows arbitrary-precision decimal computation. This comes in a substantial tradeoff in speed, but, thanks to our small problem size, the iterations are reasonably fast.

Our implementation of CFR, along with the accompanying *The Resistance*-specific functions and documentation for how to use it for other, non-*The Resistance* problems, can be found at `https://github.com/mikeambrose/ResCFR`.

VI. Results

1. LP Results

In the three-round simplified version of the game, the number of strategies is far smaller. In analyzing the game, we were able to reduce the number of spy strategies to four, all relating to behavior on the first mission: whether to pass/fail the mission if there are one/two spies on it. Thus the strategies are [(1P, 2P), (1P, 2f), (1f, 2P), (1f, 2f)]. Other than the first mission, the spies must always fail the mission if possible, either to prevent a loss or achieve a win.

The number of Resistance strategies can be reduced to 150. On the second mission, if the first mission passes, they have 3 distinct options for the second mission, and only 2 if the first mission fails. And then on the third and final mission, they have between 3 and 7 relevant strategies, depending on the histories of the missions at that point in the game.

Given these Resistance and Spy strategies, we computed the payoff matrix of the game by computing the probability each Resistance strategy would win against each Spy strategy. This is completely determined by the locations of the spies at the beginning of the game; thus, each value in the matrix is some $x/10$ where x is the number of spy configurations in which the spies win the game with this pair of strategies.

Then, we used an LP solver to find the optimal value of this zero-sum game, which was $6/10$. This was achievable with a pure strategy, where the Resistance always selected missions in a certain way, and the spies always fail the first mission[5]. Though it was interesting that this three-round variant of the game had a pure Nash equilibrium; as we

[5] Specifically, AB play round 1. ACD play round 2. If round 1 passed and round 2 failed, AB play round 3. Else, if round 1 failed and round 2 passed, AC play round 3. Also, spies could choose to always pass the first mission, rather than always failing it, and this is equally effective.

discovered, this does not extend to the full five-round game.

Though we were encouraged by the magnitude of the reduction we were able to achieve, and the solution we were able to find to the three-round game, this method did not extend to the full game. As described above, even after exploiting the available symmetries, the size of the payoff matrix was many orders of magnitude greater than any existing LP solver can handle.

2. CFR Results

The CFR algorithm was run for 5 hours on a machine with 8GiB of memory and a 3.6GHz processor, completing 4000 iterations at approximately 5 seconds per iteration. The output was two strategies which we will denote $\mathcal{S}$ and $\mathcal{R}$ (for the Spy and Resistance strategy, respectively). In order to demonstrate that the algorithm had converged, we also calculated the Spies' best response to $\mathcal{R}$, $\mathcal{S}^*_{\mathcal{R}}$.

Table 1: *Spy win rate against* $\mathcal{R}$

Strategy	$\mathcal{S}^*_{\mathcal{R}}$	$\mathcal{S}$
Win rate	0.70002	0.69914

These numbers are already very close together, with spies gaining an advantage of less than 1% by exploiting the Resistance strategy. With more computing power and time, this gap could be closed further. Furthermore, we looked at win rate by starting position (the only hidden information and the only random information).

Table 2: *Spy win rate by starting configuration*

Spies on initial mission	$\mathcal{S}^*_{\mathcal{R}}$ winrate	$\mathcal{S}$ winrate
2	0.99579	0.99484
1	0.79417	0.79345
0	0.4131	0.4119

Even in each subcategory, the deviation is very small. This gives us more confidence that our solution is almost converged. Again, given more computing power and time, we could certainly lower this gap; however, we can also be confident that the computed equilibrium is very close to a Nash equilibrium.

3. Viewing the Results

For the practical use of these results, we've published an interactive version of the tree, which is available at `https://www.ocf.io/ambrose/cs270/spies.html` and `https://www.ocf.io/ambrose/cs270/res.html` for the Spy and Resistance perspectives, respectively. These displays of the trees are simple ways for each team to follow along with the flow of the game and find the optimal mission proposal probabilities at each step. Symmetries are expanded for clarity, although early termination still occurs. Missions/spy positions "A" through "J" are referenced - these refer to the $\binom{5}{2} = 10$ possible missions, see Appendix Section 2 for an explanation. Click on a node to expand it.

VII. Discussion

The first and most obvious result is that if all players play optimally, the spies win 70% of the time. In other words, the game is biased towards the spies, which we had suspected but previously had no way of substantiating.

By examining the strategies, we see that the 5-round game is inherently more complex than the three-round variant and does have some interesting dynamics. Many decisions of which mission to propose or whether to pass or fail are neither pure nor uniform. For instance, using the tree at `https://www.ocf.io/ambrose/cs270/res.html` and looking at the mission proposal distribution after a single fail, we see that, while Resistance strongly avoid proposing missions H, I, and J (which contain both members of the first mission, so are guaranteed to fail), the rest are mixed. Furthermore, there is some aversion to picking the complement to the first mission ("A"), even though the first mission failed, since a fail in that mission leaves the Resistance with extremely little information gained. Similar interesting mixed strategies can be found at a variety of places in the tree.

The Spy strategy tends to be a bit more binary. Spies tend to pass on the two-person rounds and fail on all others. Nonetheless, the final Spy win rate is extremely close to 0.7, which speaks to some fundamental symmetry within the game. It seems likely that the final win rates would converge to (1, 0.8, 0.4) for two, one, or zero Spies on the first mission. While it is more difficult to see why this is true than in the three-round game, most likely this can be arrived at via some probabilistic argument based purely on initial position.

Another interesting result for those who play Resistance is the spy strategy of almost always passing on round 1. We were confused to see this at first, since failing on round 1 means only two more fails are necessary for the spies to win the game, putting them in a superficial position of advantage. However, the first round is only a two-person round, which means that failing reveals a lot of information about who the spies are. Spies are more likely to fail on a three-person round, which leaves a lot of mystery about their location.

Overall, the strategies here likely have limited impact on the typical game of *The Resistance*, where players generally rely on other players' mistakes to win. However, some features of commonly-played strategies, like generally passing the first mission as a Spy, might be applicable to any player of the game.

Our implementation of CFR proved capable of solving this game with reasonable accuracy in a short amount of time. In general, this approach of simplifying/reducing a game and then applying CFR on the simplified game might be useful in a variety of games. For instance, we can use this mechanism design in many games in which collusion is a valid strategy. In that respect, we could solve smaller, two-person games (colluded vs. unaffiliated) to demonstrate the effect of collusion. However, this only yields a significant speedup if this also decreases the number of relevant information sets, as our mechanism did by eliminating voting/proposed missions.

Future work could look more deeply at the found strategy to discover the source of the symmetries, as well as study generalized variants of *The Resistance* with more rounds or more players. Another variant involves 7 players with 3 spies, which would increase the size of the game by around 3 orders of magnitude - doable, but far more demanding than the current implementation. Also, *The Resistance* can be generalized to any number of rounds. It seems logical to expect that Resistance win rate should increase as the number of rounds increases, assuming mission size remains the same, since the Resistance gains more and more information about the location of the Spies.

Resistance is Futile

References

[1] Sale numbers are approximate and likely an underestimate, based on number of owners on `https://boardgamegeek.com/boardgame/41114/resistance` and analysis of another game designer/publisher `https://boardgamegeek.com/article/12917617#12917617`

[2] Bowling, Burch, Johanson, Tammelin, "Heads-up Limit Hold'em Poker is Solved," `http://ai.cs.unibas.ch/_files/teaching/fs15/ki/material/ki02-poker.pdf`

[3] Billings, Burch, Davidson, Holte, Schaeffer, Schauenberg, Szafron, "Approximating Game-Theoretic Optimal Strategies for Full-scale Poker," `http://webdocs.cs.ualberta.ca/~darse/Papers/IJCAI03.pdf`

[4] Zinkevich, Johanson, Bowling, Piccione, "Regret Minimization in Games with Incomplete Information," `http://poker.cs.ualberta.ca/publications/NIPS07-cfr.pdf`

[5] Neller, Lanctot, "An Introduction to Counterfactual Regret Minimization," `http://modelai.gettysburg.edu/2013/cfr/cfr.pdf`

[6] Yao, "A Theoretical Study of Mafia Games," `http://arxiv.org/pdf/0804.0071.pdf`

VIII. Appendix

1. Selecting $\mathcal{S}_Q$ From Mixed Strategy

Given a mixed strategy of Resistance trees $\mathcal{S}_t$, we need some way to extract our mixed strategy for each round $\mathcal{S}_Q$. Simply picking a tree is not sufficient because then the Spies will know exactly what the Resistance will do in the future and can tailor their response to that. Therefore, each $\mathcal{S}_Q$ must not leak any information about the future strategy.

First, we will establish some notation. Let t would be a Resistance pure strategy and let $t(r_1 r_2 r_3 \cdots r_k) = M_{k+1}$ be the mission proposal of t after the sequence of pass/fails $r_1, r_2, r_3, \cdots, r_k$. For instance, $t(p)$ be the choice of that strategy after one pass. Similarly, $t(pp)$ would be the choice of that strategy after two passes. For the pure strategy in **Figure 2**, $t(p) = A$, $t(pp) = A$, and $t(pf) = B$. We also must formalize what we mean by public state Q. In this mechanism, the only relevant public information is the mission chosen and the result of that mission. This eliminates the voting information and the other proposed missions; however, in this mechanism, no missions are rejected and all missions are accepted unanimously, so this information is useless. Therefore, our state can be represented as $M_1r_1/M_2r_2/\cdots$ where M_i is the mission proposed on turn i and r_i is the result (either p or f) on round i. Extending this notation, let Q_{M_1} be the first mission proposed and Q_{r_1} be the first mission proposed.

We define $\mathcal{S}_Q(M_i)$ as the probability that mission M_i is chosen and $P(t)$ as the probability that a given t is chosen in the original mixed Nash equilibrium S_t

$$\mathcal{S}_Q(M_i) = \sum_{t\in T^*(Q+M_i))=M_i} P(t) \Bigg/ \sum_{t\in T^*(Q)} P(t) \tag{1}$$

where T^* is defined as the set of all trees in which Q is a reachable state:

$$\begin{aligned} T^*(Q) = \{t \in \mathcal{S}_t | t() = Q_{M_1} \\ \cap\, t(Q_{r_1}) = Q_{M_2} \\ \cap\, t(Q_{r_1}Q_{r_2}) = Q_{M_3} \\ \cap \cdots \} \end{aligned} \tag{2}$$

and $Q + M_i$ is Q expanded with the decision of M_i on the next round (note that this is still valid and does not need r_i since the last spy decision is not needed for the Resistance strategy up to round i).

From that set, we pick the distribution of current strategies proportionally to their probabilities in the original mixed equilibrium.

This strategy does not give away any information about future strategies, since it is computed before any future strategies are chosen and is independent of future choice. Therefore, all that remains is to show that the probability of playing a tree is equivalent to its original probability in the equilibrium S_t.

Theorem: For each round k and for any mission results $R \in \{p, f\} \times k$, the probability of playing strategies drawn from $\mathcal{S}_Q$ up to round k that result in state Q^* is equal to the probability of picking a tree whose top k levels leads to the state Q^*

Proof: First, we will get an equation for the probability of picking a tree whose top k levels leads to Q^*. For arbitrary k, that probability is the probability of picking a tree in $T^*(Q^*)$

$$\sum_{t\in T^*(Q^*)} P(t) \tag{3}$$

since $T^*(Q^*)$ is, by construction, the set of all trees where Q^* is possible.

To give an intuition as to why this is true, we look at $k = 1$. The set of working information before picking a mission is empty (since no missions have been proposed yet). Therefore, our $Q = \{\}$ and, for each mission M_i,

$$\mathcal{S}_{\{\}}(M_i) = \sum_{t\in T^*(M_i)} P(t) \tag{4}$$

(there is no proportional constant since $\sum_{t\in S_t} P(t) = 1$)

Therefore, at the end of round 1, for any $Q^* = \{M_1 = M', r_1 = R_1\}$ we must show that the probability of arriving at Q^* through $\mathcal{S}_Q$ is equivalent to the probability of picking a tree that starts with M'. In other words,

$$\mathcal{S}_{\{\}}(M') = \sum_{t\in T^*(M_i)} P(t) = \sum_{t\in T^*(Q^*)} P(t) \tag{5}$$

which is directly true from the definition of $T^*(Q)$.

Now we prove the full case. We write our probability that Q^* is drawn from $\mathcal{S}_Q$s as

$$\prod_{i=0}^{k-1} \mathcal{S}_{Q_i}(Q_{M_{i+1}}) \tag{6}$$

where Q_i is the set of strategies up to round i. Expanding each S_{Q_i} gives us

$$\begin{aligned} \sum_{t\in T^*(Q_1)} P(t) \times \frac{\sum_{t\in T^*(Q_2)} P(t)}{\sum_{t\in T^*(Q_1)} P(t)} \times \cdots \\ \times \frac{\sum_{t\in T^*(Q_k)} P(t)}{\sum_{t\in T^*(Q_{k-1})} P(t)} \end{aligned} \tag{7}$$

Alternate terms cancel, giving us

$$\sum_{t\in T*(Q_k)} P(t) \tag{8}$$

where Q_k is the same as our final state, Q^*. This is exactly the same as our alternate definition, so the lemma holds and our method of generating $\mathcal{S}_Q$ is equivalent to the Nash equilibrium $\mathcal{S}_t$.

2. Enumeration of Position Sets

In the visualizer (mentioned in section *VI.3*), missions and spy positions are referred to by letters A through J. These refer to sets of positions as follows (with players numbered 1 through 5 around the table from an arbitrary start index of 1):

For spy positions and 2-person missions:

Table 3: *Mission Distribution*

Name	Players On Mission
A	1, 2
B	1, 3
C	1, 4
D	1, 5
E	2, 3
F	2, 4
G	2, 5
H	3, 4
I	3, 5
J	4, 5

For the 3-person missions, take the complements of above.