Best Strategies in Repeated One Vs. Many Games: Dead by Daylight

Toluwase Afolayan
Department of Physics
tafol20@bu.edu

Fasial Mushayt
Department of Computer Science
f.mushayt@gmail.com

Abstract

The One Vs. Many archetypes are one that shows up in many interactions in modern society. Examples being Parents vs. their children, Employees Vs. Employers, Teacher Vs. Student and Large distributors Vs. Small Business. In an effort to study a formulation of One Vs. Many formats, a simulation was created to encapsulate a particular One Vs. Many games, called Dead by Daylight, through N-person Extensive Game Decisions Tree. Dead by Daylight is a One Vs. Many games where a single Killer hunts four survivors who are trying to escape. The decision trees that define this game simulate different scenarios in the game, and different players' action sets within the game. We define five player strategies and pit them against each other in repeated games. In the end, there didn't seem to be a strategy that generally outperformed the rest. The One Vs. Many Scenarios are particularly interesting because of the way it mimics the betray and cooperation group interplay of modern everyday life.

Introduction

In modern society there are many circumstances where socio-political and socio-economic interactions organize themselves into particular thematic circumstances. Society tends towards small groups of entities and people in positions of power and larger groups of people in positions of some type of servitude or subjugation. A great example of this concept is the relationship between an Employer and their team of employees. The employer might be a slew of managerial entities, but the relationship can often be represented in real life as a manager Vs. a team of employees. In this scenario, the manager can have actions such as controlling how much employees can work, how much they will be paid, and what roles they will have in furthering the goals of the manager, and thereby hopefully the company as a whole. For the employees, their actions are defined in two fairly distinct sections: their interactions with the manager, and their interactions with their fellow employees. For example, it is entirely possible to see how a employee who interacts poorly with their peers will potentially not be able to excel as much as they might if their relationships with their fellow employees were better. So in reality for the employee, they must react and act in relation to what actions the employer takes, but also in order to maximize their benefit it is in their interest to work well with their fellow employees also. The issue arises when maximizing benefit with employers and maximizing benefit with fellow employees, become at odds. This concept of maximizing odds might be represented by a call by employees to assemble themselves into a union being at odds with a manager's want for employees to not cooperate with each other in this way.

This employee vs. employer scenario is one of many examples that outlines this concept, amongst others like farmer Vs. Farmhands, class-action lawsuits filed against large companies by consumers, or workers, etc. The theme is a larger more powerful entity maximizing its benefits against a group of many smaller players working individually with and against each other and against the singular entity.

In-order to properly analyze the interaction just described, we have to formalize this into a game representation that encapsulates a few different characteristics well, namely; the interactions between a single player of the many sets and the one entity, and the other players of the many sets. Initially we tried to set up a zero-sum game between a “One” entity and a “Many” entity. This very quickly became an inoperable solution. The zero-sum game did not encapsulate the interactions between the many players very well if at all so we sought out a different solution entirely. This led us to another game formulation, the N-Person Extensive Game.

N-Person Extensive Games

In-order to properly simulate the “One Vs. Many” scenarios, we needed a system that can show how a single many player acts in relation to other single many players as well as in relation to the one player. We wanted a system where a single many player can take an action, amongst its fellow many players that they will react to, and likewise react to the decisions of other players in the many set.Then depending on these actions sets, players from the many set can come into contact with the One player. It appeared that the best way to simulate this action and reaction system is to set up a system that takes into consideration the series of actions taken by both the many players and the single players. We decided that due to its sequential nature, the extensive N-person game would be the best way to form the game we were looking for. In particular we would build our one vs. many scenario using a multitude of extensive game decision trees.

Here, each action taken by the players of the game would result with a set of payoffs, they would receive scores for their actions, and the consequences of these actions, and these trees would continue until some final state is reached where a final payoff can be calculated.

Given this idea we looked for an interesting example that was more feasible to implement as a simulation. Something with a manageable set of “One” and “Many” actions that can be simulated. This led us to a video game called Dead by Daylight, a One Vs. Four survival horror video games developed by the game development company called Behaviour Interactive. As we will see, the game very well fits the archetype we developed.

Dead by Daylight

As aforementioned, Dead by Daylight is a One Vs. Four, survival horror game that pits a single Killer player against a set of Four Hunter players. The goal of the Killer player is to hunt down and Kill each survivor player. The Survivor players goal is to avoid being killed whilst they look for and repair a total of five generators that are placed randomly round a fairly expansive map filled with rooms, corridors, traps and other things. After finishing these five generators the Survivors have the ability to open a door that, once open, can allow them to escape the map and achieve victory.

The killer kills in essentially one way. The killer roams the map looking for survivors to kill, if the killer finds a survivor, they chase them down, and hit them, injuring them. If the killer is able to hit them twice in a row, the killer will knock the player to the ground. At this point, the killer's next objective is to take the downed survivor and impale them on a sacrificial hook. At this point a timer will begin, at the end of this timer, the survivor who has been hooked will die. It should be noted that in the injured, downed or hooked states, another survivor must come and assist the afflicted survivor to get them out of this state, either by healing, helping up or unhooking them respectively. If this is the first time a survivor has been hooked by the killer they will be able to attempt to unhook themselves 3 times before being the hook timer is cut in half, this unhook attempt has a very low chance of succeeding. Besides fixing generators there are a multitude of other tasks that a player can conduct that have far lower total game pertinence but can result in other smaller rewards, like medkits to self-heal, flashlights, to stun the killer, or toolkits to destroy hooks or fix generators faster.

For the survivor the game is a loop of essentially fixing generators, and unhooking teammates while avoiding getting hit by the Killer player. For the Killer the game revolves around checking generators for survivor players to attack and watching already hooked or injured players, and making sure they die. Another notable feature is that if all but one player is dead, then a trapdoor that is placed randomly on the map opens, and can be used by the last player to escape and win even if all the generators aren't fixed or the door isn't open. Now we will discuss exactly how the Dead by Daylight extensive game is formed.

Extensive Game formulation

For the game's extensive game a series of extensive decision trees were built to model the games main features. For ease of understanding, and purposes of practicality many of the finer details of the game were extrapolated into a system of probabilities that will be soon outlined. For the game we defined four trees; The Killer, Survivor, Chase, and Hook tree. We have broken the actions up in a regimented understandable way. Starting from the killer tree. The killer tree(fig. 1) is the simplest of the four, the killer just picks one of 7 generators in the game, and visits it, if it finds at least one survivor it enters the chase tree with this survivor. If it finds no survivors, the next round the killer will check a different generator and so on. At the end of their turn, if the killer has not found anyone at a generator, they will search for each survivor. This is done to simulate the survivor randomly coming in contact with the killer away from a generator or a hook, something we chose not to simulate in real time.

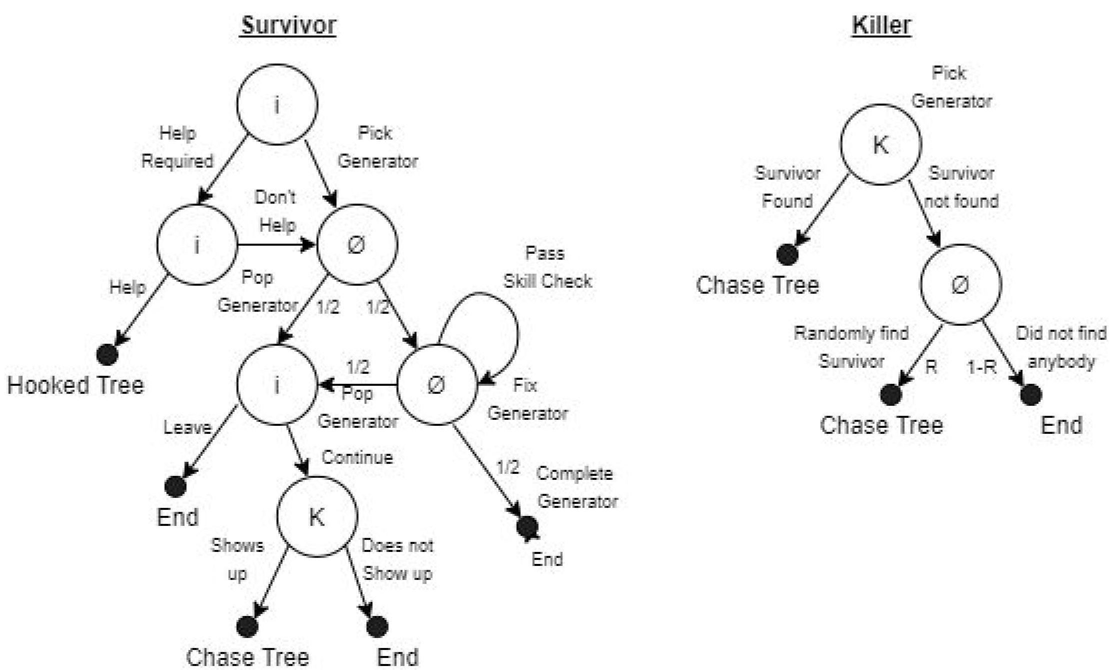


Figure 1: The Survivor and Killer Decision trees

For the Survivor Tree(fig. 1), the players will either also pick a generator to go to and fix or they will attempt to heal a fellow survivor that has been hit by the killer depending on some strategic probability. Up to three survivors can fix a single generator at once. In the game there is a concept called a skill check, whilst fixing a generator the player is prompted to press a button in a certain timeframe, if they are unable to press the button on time that survivor “pops” the generator. This creates a sound that the killer hears, and lets him know of at least one generator where the survivors will be. At this point the killer will likely go to the generator if he is not busy with another action. Before the killer shows up the survivor who pops the generator will have an opportunity to either take their chances and continue fixing the generator, or they will leave and hide to avoid being caught. At this point if the survivor stays at the generator and the killer shows up, the survivor and at most one other survivor who is at the generator will enter the chase tree to simulate the chase process of the killer.

The chase tree(fig. 2) is the decision tree entered by the killer and at least one survivor. It is meant to simulate the action of a killer chasing and attempting to down a survivor while the survivor attempts to escape. The chase tree plays out as follows; The survivor who is being chased has the option of either attempting to use an obstacle to escape from the killer, a stun with a flashlight to escape or simply attempting to escape by running. Each strategy will have a probability of success depending on whether they are by themselves or with another survivor, and if you're with another survivor depending on the action they choose your chances of survival will be lowered or raised. For example, stunning the killer lowers your personal chance of escaping but raises your fellow survivors' chance of escaping. Each survivor will have a personal probability of picking any one strategy over another defined by their strategy type. If the players succeed in escaping, they go back into the survivor tree to wait for another round. If they fail they are hit by the killer and they re-enter the chase tree injured. If they succeed now they escape but are injured, and their R value is higher,as in the real game the killer can fall the trail of blood they make as they run away. If they fail in this second round they are downed and will be hooked, in order to be sacrificed. Once hooked the survivor will enter the final tree we will discuss, the hooked tree.

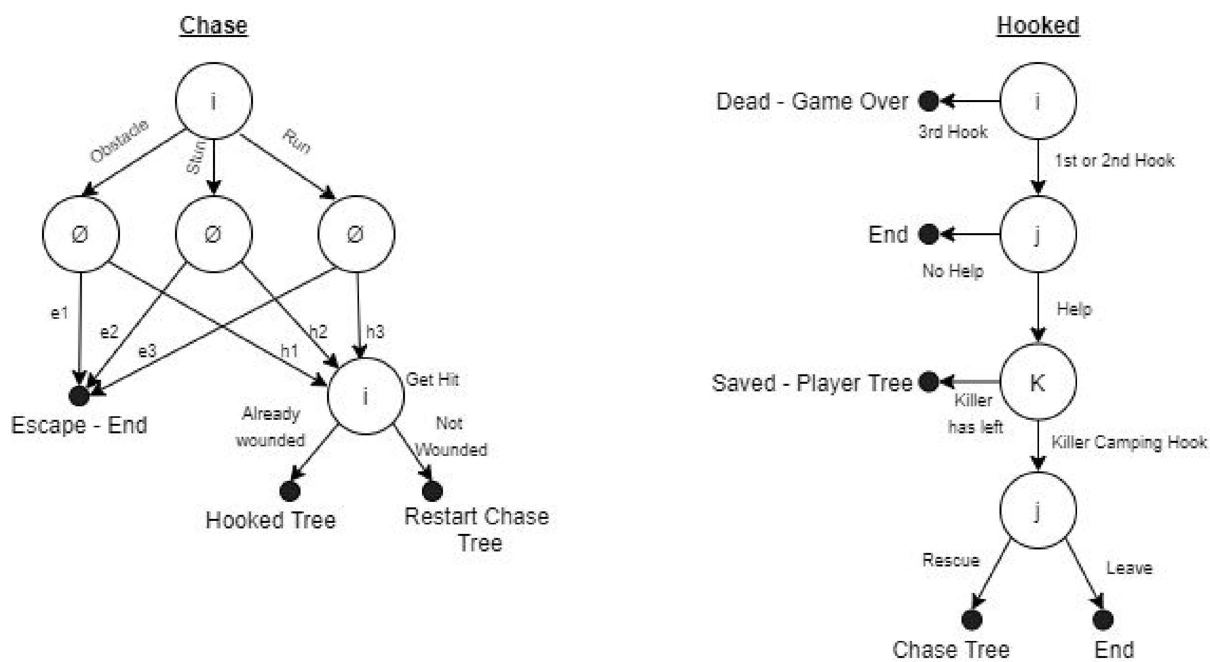


Figure 2: The Chase and Hooked Decision Trees

The Hooked Tree (Fig. 2) is only initiated when the killer hooks a player and is the tree that is played when they are hooked. It functions as follows; The player can be hooked for at most 3 turns, after which they will be sacrificed. While hooked another survivor has a chance to come and save them this is dependent on this survivor's strategies. If the killer decides to camp, i.e.; stay at the hook, when the fellow survivor comes to help, both survivors will be placed into a chase decision tree with the killer. For the survivors there are 5 unique strategy sets that each player will play with. They are outlined in the table below.

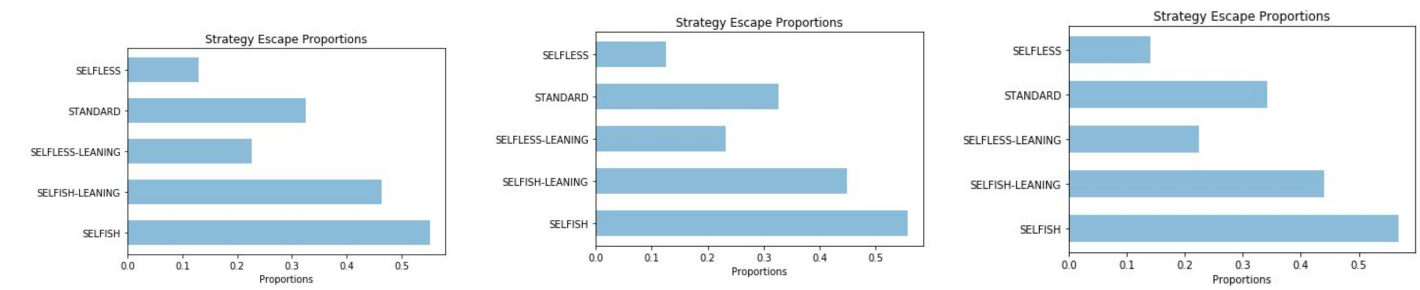
Strategies	Standard	Selfless	Selfless-Leaning	Selfish	Selfish-Leaning
Run	1/3	1/10	2/10	7/10	5/10
Obstacle	1/3	7/10	5/10	1/10	2/10
Stun	1/3	2/10	3/10	2/10	3/10
Rescue	1/2	95/100	7/10	5/100	3/10
Leave	1/2	5/100	3/10	95/100	7/10
Continue	1/2	8/10	6/10	2/10	4/10
Leave	1/2	2/10	4/10	8/10	6/10
Heal	1/2	1	7/10	0	3/10
Ignore	1/2	0	3/10	1	7/10

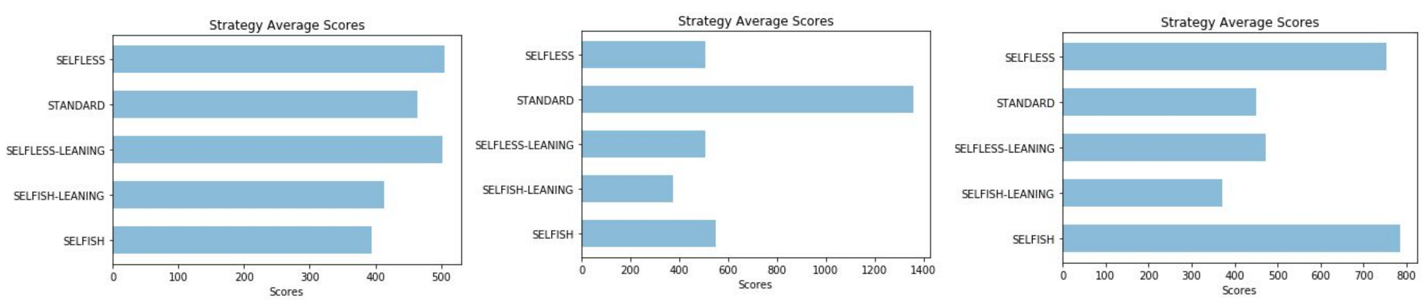
Table 1: The strategies simulated in this game and the strategies behind them.

The game is simulated by running through these game trees until one of two things happens, either all players are dead or five generators have been repaired, at which point, two more turns will pass before the door opens and all still free survivors escape. At each action that the player makes, and if the players win the round, payoffs will be rewarded and each player will gain some score which at the end, if they end up surviving, also scales with the amount of other survivors that also escaped. This system was made to match the point award system of the game.

Results

We run this game simulation to completion 100 times for all combinations of survivors to see how each survivor strategy type fairs in the games. This data is recorded, from this data we calculated three values which were plotted. These are Strategy Escape proportions (Fig. 3), Average Strategy Scores (Fig. 4), and Strategy Cumulative Scores (Fig. 5). Note each column in this case represents a full pass through of the simulation. Since the outcome is not deterministic, these are provided to compare and discuss.





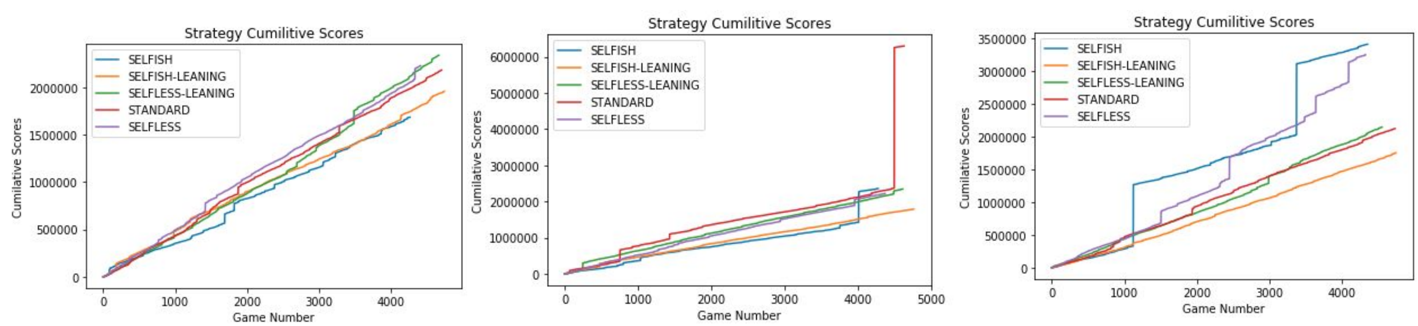


Figure 3 to 5 from top to bottom, Escape proportions, Average Strategy Scores, and Strategy Cumulative Scores
Figure a to c from left to right, seperate run throughs of simulation

Analysis & Discussion

At a first glance, the escape proportions shown in Figure3 might indicate that the game is rather deterministic. All three parts of the figure are almost identical. When we look at the scores in Figure4 however, the situation is entirely different. We had initially thought that since the escape proportions seem to converge to a common distribution, then the scores should also converge. That was not the case. Scores would vary widely from run to run. Although most of them resemble Figure4a and Figure4c, where two or more strategies will be neck to neck in scores, sometimes we will get extreme scenarios like that of Figure4b where one strategy will jump way further ahead or will fall significantly behind. This was somewhat bizarre at first as all players have the same scoring scheme. If one strategy was truly superior to the others, then we should see it consistently outscoring the rest. That never occurs however, it is only occasionally a random strategy will become superior to the rest.

At this point we looked at the scores from another perspective: the strategies' accumulated scores over time. These plots showed us the bigger picture. We can see in the case of Figure5a that all the strategies tend to have a similar upward trend, with Selfless, Selfless-leaning, and Standard doing slightly better than Selfish and Selfish-leaning. This makes perfect sense considering what we saw in Figure4a. Now, Figure5b also seems to show this trend, but suddenly has a huge spike in Standard's score near the end. Similarly, Figure5c shows spikes in Selfless's and Selfish's scores, yet they still follow the trend otherwise. As if the lines remain parallel to one another despite the spikes. This leads us to believe that none of the strategies are actually superior to the rest in this case. Due to the probabilistic nature of this simulation, it is likely that some players might just randomly and extremely outperform others, which holds true to the nature of this video game.

Now that we know this, it is interesting to go back and look at the strategies and their escape proportions. One in particular; Selfless. Selfless players seem to consistently have a very low chance of survival, yet they still perform great score wise! This seems counterintuitive at first, but since payoffs to a player are given proportional to their score and number of survivors at the end, the Selfless player can capitalize on that by always helping others, increasing his own score, and ensuring as many people survive as possible. Kind of a wholesome outcome.

Conclusions

The One Vs. Many games are important because they contain facets of many different parts of modern life. In our game we formulated a version of this One Vs. Many that play with many fantastical elements, but at its core, it promotes similar themes. In order for survivors to succeed and maximize their scores, they have to perform group actions. Whether this is unhooking teammates, healing, working on generators together, or helping each other to escape from the Killer teamwork is very useful. In the same vein there are many situations in which abandoning a teammate can increase one's chances of coming out ahead in a particular situation such as escaping from a killer in one circumstance at the expense of your teammate, but such an action might make later scenarios more difficult to traverse. This is an important concept because it is something that seems to hint at a foundational theme of groups that engage in prolonged interaction. This idea of group betrayal and cooperation, in the face of an outside entity. This kind of interaction is what marked the evolutionary traits of our ancestors and allowed them to form into groups and family units.

In a modern world defined by the strife of hyper-capitalism, it might benefit to study One Vs. Many scenarios can possibly give insight into how to traverse a very complex and unique situation that we find ourselves in constantly.

Future Work

Concepts that we thought would be interesting to expand upon in the future definitely surround the python implementation of this simulation. In its current iteration the game was implemented with a fairly limited set of actions and states of being for the players, many interesting and dynamic pieces of the gameplay loop were folded into probability checks or ignored for the purpose of a more straightforward and streamlined game. To this end, analyzing the game would be greatly helped by a fleshing out of some of the core pieces. In particular giving more agency to the Killer might make the interactions between survivor and killers even more nuanced. Adding items, like the medkits mentioned in the game summary, will diversify the action set of the survivors making game scenarios more interesting.

The main thing we would have implemented next however was a set of more involved Survivor strategies. In particular strategies that react to and adapt to the actions of other players. This would have best encapsulated the concept of the "many" players inter-play. For an example survivor that remembered that another survivor popped a generator, and as such chose not to repair the same generators as that player again. In such a circumstance, if the killer finds that player, they might be less likely to have someone with them to help them out. This would allow us to better see how the interactions between players affect how players fare in the game as a whole. Also notable, we read a research paper about using bandit learning, a machine learning algorithm to train a player strategy

Works Cited

Bravo, Mario, et al. "Bandit Learning in Concave N-Person Games." *NIPS*, papers.nips.cc/paper/7809-bandit-learning-in-concave-n-person-games.pdf.

Shapley, L.S. "N-Person Game Theory." *Rand*, RAND, 2008, www.rand.org/content/dam/rand/pubs/papers/2008/P3752.pdf.