Ad Hoc Vehicle Platoon Formation

Robert W. Thomas
Department of Computer Science and Engineering
University of South Carolina
Columbia, SC, USA
thoma525@email.sc.edu

José M. Vidal
Department of Computer Science and Engineering
University of South Carolina
Columbia, SC, USA
vidal@sc.edu

Abstract— **Road traffic is something most people experience daily. Traffic congestion is something everyone involved seeks to avoid, but often cannot. There are many facets of research seeking to alleviate or solve traffic problems. Platooning is a method that could help. Platooning is a technique where two or more cars drive in close proximity, one behind the other. To date, platoon formation research has largely focused on centrally orchestrated planning. Allowing dynamic formation of platoons seems a natural means to increase adoption and participation, but a standard way of negotiating the platoon is needed. This paper defines an Ad Hoc Platoon formation game that meets that need and evaluates potential strategies players could employ.**

Keywords— Multi-Agent System, Autonomous Vehicles, Platooning

I. INTRODUCTION

Platooning is a technique where two or more vehicles drive in close proximity, one behind the other. This reduces the amount of road space taken by the participating vehicles, increasing the effective capacity of the road network [1]. Increasing road network capacity naturally alleviates congestion. Platooning also allows the platoon members other than the lead vehicle to draft the vehicle in front of them, increasing fuel efficiency. While Platooning is unrealistic in terms of safety with humans behind the wheel or in adverse weather conditions, advances in connected and autonomous vehicles continue to move closer to resolving vehicle dependency on human drivers [2].

When vehicles move in platoons, they increase the density of traffic, thereby increasing throughput if travel speed remains comparable. Modelling and simulation suggests that in some scenarios, road throughput could be increased by more than double [3]. Most approaches to platoon formation however rely on central orchestration to dictate how platoons are to be formed. This represents a barrier to entry requiring an extra burden on the end-user to register with such a system or systems for areas they travel. Users may also have to alter their travel schedule to accommodate platoon formation [4]. Additionally, determining what platoons should form and when is computationally expensive and would not scale well to meet the demands of a modern urban environment. Computation costs can be reduced by filtering out platoon assignments that cannot be achieved due to geographic or temporal separation in order to reduce the number of participants to evaluate [5], however an urban environment with a large penetration rate of platoon-enabled vehicles may still present a challenge.

An ad hoc means to form platoons could alleviate the barriers to entry described above. Users would not necessarily have to register with any system ahead of time. They need only to follow an accepted protocol for negotiating platoon formation. The computational cost of determining platoon arrangements at scale is also mitigated as the computation is distributed across all platoon enabled vehicles. Ad hoc platoon formation affords the benefits of platooning without the infrastructure constraints of a central orchestration authority.

Research on ad hoc platoon formation is more sparse than central planning. It is sometimes mentioned as a comparison to centralized methods but only in simplistic terms such as ad hoc platoons forming only when two vehicles are directly in-line with each other [4]. Recent research has looked at algorithms for ad hoc platoon formation with nearby connected vehicles. The proposed platoon members elect a lead vehicle based on route information and driving characteristics such as consistent speed and lane choice [6]. This approach however may penalize “good behavior” since a vehicle exhibiting desirable driving qualities is more likely to be set as the platoon leader, forgoing fuel efficiency benefits. Furthermore, nefarious users might even try to misrepresent their behavior to reduce their chance of being selected as platoon leader. Even if vehicles exhibiting dubious behavior are filtered out using a behavior model to establish trust among peers [7], there is still the problem of rewarding “good enough” behavior over exemplar driving behavior.

For ad hoc platoon formation to be possible, an accepted protocol for negotiating platoon formation among peers is needed. To be successful, that formation method needs to incentivize good behavior in a fair manner. A standard game

978-1-7281-0137-8/19/$31.00

with pre-defined rules known to all participants could serve as the platoon negotiation method.

II. BACKGROUND

In game theory, a game consists of three components. They are players, actions the players may take, and a payoff matrix. The payoff matrix determines the utility, or reward, that each player gets based on the combination of actions chosen by all the players. Players select actions based on their strategy. Solutions to a game assign a strategy to each player. The outcome of a game is the utility received by each player after actually playing [14].

One may view the Ad Hoc Platoon formation problem as an iterative game. Everyday vehicles encounter one another. Inevitably, vehicles with similar commutes have repeat encounters on occasion. To make the problem tractable, we assume that two vehicles travel the same road at the same time on their daily commute, thus encountering each other every day in exactly the same way. We also assume that road conditions and weather are ideal. Vehicles may drive the road independently, or they may link in order to drive the road as a Platoon with one vehicle following closely behind the other. The tail in the link receives a benefit in fuel savings due to experiencing reduced air drag while in the Platoon. Either vehicle may initiate or terminate a link at any time during the journey. The vehicles are not allowed to "tailgate" for the fuel benefit when not linked. The journey is repeated iteratively, e.g. once per day. Initiating a link results in an iteration of the repeated game.

When either vehicle initiates a link, each may choose to Lead or Follow. To Lead the Platoon means taking the head position of the link. To Follow means joining the Platoon in the tail of the link. Being at the head or tail of the link impacts the reward received by each vehicle. The formation of a link and the position of each vehicle in it depends on the choice made by both vehicles.

If both choose to Lead, which is head and which is tail is determined randomly by a third player "chance". The player chance represents a coin flip to decide the leader vice a literal player. One can expect to follow 50% of the time in iterations where both choose Lead. This creates the need for two payoff matrices depending on which vehicle would lead in the event of both choosing Lead. If neither vehicle chooses lead, neither vehicle will get any fuel savings benefit. The payoff matrices are depicted below in tables 1 and 2. Fig. 1 depicts the extended form game.

The egalitarian solution for the Ad Hoc Platoon Formation game is for both vehicles to always choose Lead, or tit-for-tat. The Pareto optimal and social welfare solution is both vehicles to always choose lead, tit-for-tat, or one always follows and the other always leads.

This game may be expanded to three or more players by processing multiple link requests per iteration. That is, multiple two-player instances are played per turn until every available pair combination of players has been tried. Once two vehicles link, they act as one unit with the head negotiating future links to allow other vehicles to join the Platoon or have all members of the Platoon join another Platoon. If a vehicle joins a Platoon as a follower, it stops listening for links requests and is no longer considered available. New link requests would go to the lead vehicle. A vehicle may only join a Platoon as a follower once per iteration. The iteration concludes once all vehicles have joined a Platoon or attempted to join one with all other vehicles listening for link requests.

TABLE I. PAYOFF MATRIX CASE 1

Car 1/Car 2	**Lead**	**Follow**
Lead	**0,1**	0,1
Follow	1,0	0,0

TABLE II. PAYOFF MATRIX CASE 2

Car 1/Car 2	**Lead**	**Follow**
Lead	**1,0**	0,1
Follow	1,0	0,0

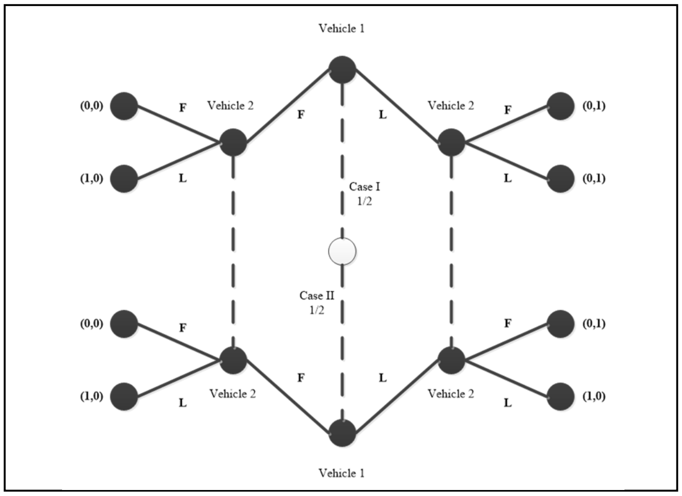

Fig. 1. Ad Hoc Platoon Game Extended Form

This game is similar to the Iterated Prisoner's Dilemma (IPD) [8]. Individual players are free to select their own strategy for negotiating a platoon. Players seeking to maximize their score may be tempted to adopt a strategy of always choosing to follow. Just like the double defection in IPD however, neither player gets what they want, to the detriment of both. The incentive to lead the platoon is more subtle than the quick payoff from following. The incentive to choose lead for a greedy agent is the promise of future rewards from a stable, cooperative community that can offer greater benefits in the long run.

How effective different player strategies are in achieving benefits for the player and for the traffic system overall is a question this paper addresses. Given the similarities between the Ad Hoc Platoon game and IPD, strategies developed for one can be evaluated in terms of the other. The approach uses a round-robin style tournament to determine how well IPD strategies perform against each other while playing the Ad Hoc Platoon game.

A. Axelrod Tournament

In the 1980s, Robert Axelrod in his study of the IPD game invited fellow researchers to submit their best strategies. He then conducted a round-robin tournament amongst all entries. The results proved interesting as cooperative strategies performed better, despite not being a dominant strategy from a game theory perspective [8]. IPD continues to be a research area of interest for many.

Axelrod-Python is a library actively maintained on GitHub that supports IPD research. It provides a framework for researchers to setup their own tournaments with access to over 200 strategies submitted by various contributors. The library also offers a number of built-in methods to support analysis and visualization of results [9].

B. Basic Player Strategies

Five basic prisoner dilemma strategies implemented in the Axelrod library were evaluated in terms of the Ad Hoc Platoon game. The strategies used are Cooperator, Defector, Tit For Tat, Grudger, and Random (0.5). Tit For Tat, Grudger, and Random were part of Axelrod's original tournament, with Tit For Tat wining overall [10]. Cooperator and Defector were part of a 2012 variant of Axelrod's Tournament run by Alexander Steward and Joshua Plotikin [9].

Cooperator always cooperates. In terms of the Ad Hoc Platoon game, this equates to always choosing Lead. This strategy is beneficial on a system level as it ensures a Pareto optimal solution as it guarantees a Platoon will form, regardless of opposing strategy. It is altruistic in nature as it ensures the opponent will get the Platoon benefits. Against another cooperative strategy, Cooperator will receive a payout 50% of the time on average, depending on chance.

Defector always defects. For single play or fixed duration iterative games, it is the dominant strategy for greedy players playing the Prisoner's Dilemma game [8]. In terms of the Ad Hoc Platoon game it equates to always choosing follow. This would yield high individual scores when competing with cooperative strategies, but result in no benefits for either player when competing with other greedy strategies.

Tit For Tat cooperates for the first turn, and then mimics its opponent's move on the next turn. It is able to benefit from cooperating with cooperative opponents while punishing greedy ones, e.g., Defector. Despite not being theoretically robust, it performs well in application as indicated by winning Axelrod's early tournaments [11].

Grudger starts out cooperative but holds a grudge if its opponent defects. That is, it cooperates until its opponent defects. After a defection by its opponent, Grudger plays defect for the remainder of the match. In Ad Hoc Platoon game terms, Grudger chooses Lead as long as its opponent does the same. If the opponent chooses Follow on any move, Grudger will choose Follow on the next move and continue to do so until the match ends.

Random randomly selects between cooperate or defect, Lead or Follow in Ad Hoc Platoon terms. Random (0.5) has a 50% of going either way. When paired against a cooperative opponent, it will do better than its opponent, though not as good as Defector. When going against a greedy opponent, Random will punish its opponent more so than a more cooperative strategy, but not as much as a pure Defector.

C. Moran Process

A Moran process or model is used to study variety over time for a fixed-size population. It can be used to model how a population might become either more heterogeneous or more homogenous over time. At each time step, a random member of the population is chosen to reproduce, creating a copy of itself. Additionally a random member is also chosen to die in order to keep the population size constant. The selection for reproduction is based on some measure of fitness, giving different groups a varying probability of being chosen as the reproducer. Monitored over time, the population attributes shift, presumably in favor of the more fit members of the population as they are more likely to be the ones reproducing on any given turn [12].

The Axelrod Python library includes methods for applying the Moran process to a population of players with different strategies. In this game variant, after each round between two players, one player is chosen to reproduce. The chance of selection for each player is proportional to their score for the round. Then one of the players is chosen at random to be replaced. After the reproduction-replacement steps complete, one of the players may or may not have effectively changed strategies. The more successful the strategy, the more likely it will reproduce and conversely, the more poorly a strategy performs in a given round, the more likely it is to succumb to natural selection. The basic Moran process game ends when all players are using the same strategy [9].

III. METHODOLOGY

The methodology used to investigate the effectiveness of different strategies at playing the Ad Hoc Platoon game was to play the strategies against one another in a round-robin style tournament. Two different tournament types were used. First, strategies competed in a traditional tournament with every player facing every other player, maintaining their original strategy from one round to the next. Second, strategies played against each other using the Moran process to emulate natural selection. This allowed players to switch strategies between iterations based on how their strategy faired relative to the other strategies.

The Ad Hoc Platoon tournament was created by modifying the Axelrod library [9]. Two key changes were made. The payoff matrix was modified to reflect the Ad Hoc Platoon Game matrices in tables 1 and 2 and the opportunity for the non-player "chance" to select which matrix would be used before each individual round was added. Which payoff matrix was in play was unknown to players until after the round was played.

The setup for the first tournament consisted of fifty players with each individual match going for ten rounds and the overall tournament going for ten repetitions. The probability of a match ending early was zero. Noise, the chance that a player's intended action is not the move played, was also set to zero.

The tournament players all used one of the basic strategies described above. Because the mix of strategies used in the tournament could affect the overall outcome, all combinations were run and the summary results used for analysis. The first tournament was run 316,251 times to account for all distinct combinations of the five basic strategies over the fifty players. Note that since all players play each other in individual matches for the round-robin tournament, the order of players using each strategy does not matter, only the number of players using each strategy. As a result, there are only 316,251 possible combinations vice 5^50. The built-in Python function combinations_with_replacement (p,r) was used to generate the player list for each tournament instance. It produces r-length tuples in sorted order with repeated elements from the element pool p [13].

Summary data across all runs was collected to facilitate overall analysis. Summary data points collected, by strategy, were total score, maximum individual score, and specific tournament run or runs where the maximum individual score was achieved. Maximum system score, the summation of player scores, and the runs where the maximum system score was achieved were also tracked. For the calculations of both individual and system scores, the average performance of individual players across the ten repetitions of each tournament instance was used. This was done to discount potential effects of any player going on a "winning streak" with regard to which payoff matrix was selected by the non-player chance.

The second tournament type used applied the Moran process. The setup for second tournament also consisted of fifty players with individual matches going for ten rounds. The probability of a match ending early and noise were set to zero. The mutation rate, the chance that a replicated player will switch types, was also set to zero. The initial strategy of each player was one of the five basic strategies described above. The second tournament was repeated 10 times. Each time the initial distribution of strategies was even at 10 players each. The number of wins each strategy achieved and the average number of iterations needed to win were recorded.

IV. RESULTS

The results show that while defector achieved the highest individual score for a single run, more cooperative strategies, including random, outperformed it. Overall, the summation average scores of each strategy across all 316,251 runs, rounded to the nearest whole number were as follows. Cooperator amassed a score of 581,114,366. Defector achieved a score of 439,056,356. Tit For Tat scored 610,164, 354. Grudger had the best score overall of 632,775,776, and Random scored 532,654,136.

The maximum scores achieved by strategy and tournament rounds in which they were achieved are as follows. The best average score for a Cooperator on a single run was 260.0 achieved on run 1,354 which consisted of 39 Cooperators and 11 Grudgers. The best average score for a Defector was a perfect 490 on run number 2 which was one Defector playing with 49 Cooperators. The top score for Tit For Tat was 259.4 which it achieved twice, in runs 51,924 and 100,650. The first run had 19 Cooperators, 18 Tit For Tat, and 13 Grudgers. The second had 13 Cooperators, 22 Tit For Tat, and 15 Grudgers. The best average score for a Grudger in a single run was 261.9 achieved on run 20,430. That run had 26 Cooperators, a single Tit For Tat, 20 Grudgers, and three Randoms. The best score for Random was 368.9 on run 15 which consisted of 48 Cooperators and two Randoms.

A perfect system score of 12,250 was achieved 1,328 times. A perfect system score occurs when a platoon is successfully formed in every single match. Across all the runs that achieved the maximum system score, the total number of representatives from each strategy is depicted in Fig. 2 below.

The results of the second tournament again show Grudger as the best strategy. Of the ten runs, Grudger won five. The other five runs were split between three other strategies. Cooperator and Defector won two each and Tit For Tat won once. Random was the only strategy that failed to win a run of the second tournament.

The average number of runs needed to win a round of the Moran Process tournament varied by strategy. Grudger on average won in 1,692 rounds with a high of 5,432 and a low of 504. Grudger had both the quickest and the slowest victory. Defector took on average 851 with a high of 1,058 and a low of 644 rounds. Tit For Tat's single win was accomplished in 973 turns. Cooperator took the longest to when on average at 2,767 rounds with a high of 2,943 and a low of 2,591.

V. CONCLUSION AND FUTURE WORK

The results support that cooperative strategies are more beneficial in the long run. This is true for both individuals and the system as a whole. Of the five strategies, the three cooperative strategies performed the best as individuals. Defector, the always selfish strategy, performed the worst overall, far behind even Random. This happened despite the fact that it had the highest score of any strategy in a single run. As a system, cooperation was once again the key to success. Overall, the optimal system score was achieved when only cooperative strategies were in use. The only exceptions being when there was only 1 player using a "non-cooperative" strategy. If more than one player had a non-cooperative strategy, that is Defector or Random, the maximum system score was unobtainable. Cooperator being the most common

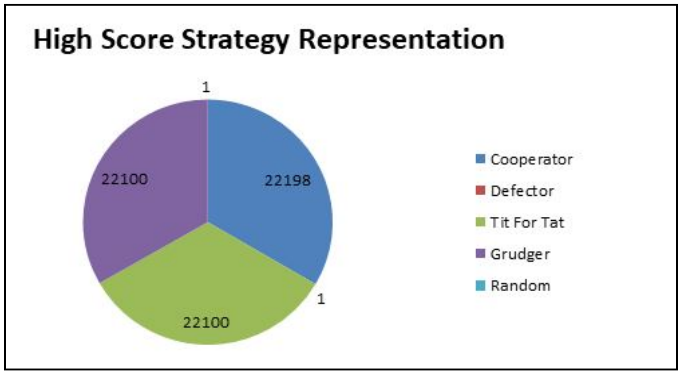

Fig. 2. High Score Representation by Strategy

strategy in system optimal runs is not surprising as it guarantees platoons with any opposing strategy while Grudger and Tit For Tat do not.

The manner in which the cooperative strategies react to selfish and neutral strategies like Defector and Random respectively determines how they rank individually and in terms of system benefit. Ironically, while cooperative strategies dominated overall, the more cooperative a strategy, the poorer it faired in comparison other cooperative strategies in individual scoring. Grudger, the least cooperative of the cooperative strategies, won overall as an individual strategy. This was because it cooperates with Cooperator and Tit For Tat while defecting against Defector and Random after being defected against. Tit For Tat would play the same as Grudger against the Defector but would try to cooperate more with Random. Cooperator, by always cooperating no matter what would only score against Random when Random played Lead and chance picked Cooperator as the follower. In the end, how the cooperative strategies played against Random is what distinguished their individual scores from each other. In terms of system benefit, how the cooperative strategies played against both Defector and Random made the difference. Cooperator was the most represented as it was able to guarantee platoon formation in the presence of one Defector and again against one Random. This allowed Cooperator to edge out Grudger and Tit For Tat which cannot guarantee platoon formation with Defector or Random.

Given the choice of one of the five strategies evaluated, Grudger seems the most preferable long term choice. A community of Grudgers, while being highly cooperative, would punish selfish behavior, effectively exiling non-cooperative players. With that in mind, a system-wide adoption of that strategy would produce a solution that is both Pareto optimal and provides equitable distribution of Platoon benefits.

Future work will focus on developing simulations to quantify potential fuel and time efficiencies of Ad Hoc Platooning compared to centrally orchestrated methods. How much the penetration rate of platoon enabled vehicles affects potential benefits will also be studied. Additional topics such as preventing the intrusion of non-participating drivers into platoons and application of the methodology presented in this paper to other problems in this domain could also be considered.

REFERENCES

1. Kavathekar, P., & Chen, Y. (2011). Vehicle platooning: A brief survey and categorization. ASME International Design Engineering Techincal Conferences and Computers and Information in Engineering Conference (pp. 829-845). American Society of Mechanical Engineers.
2. Koopman, P., & Wagner, M. (2016). Challenges in autonomous vehicle testing and validation. SAE International Journal of Transportation Safety 4.1, 15-24.
3. Lioris, J., Pedarsani, R., Tascikaraoglu, F. Y., & Varaiya, P. (2016). Doubling throughput in urban roads by platooning. IFAC-PapersOnLine 49.3, pp. 49-54.
4. Sokolov, V., Larson, J., Munson, T., Auld, J., & Karbowski, D. (2017). Platoon formation maximization through centralized routing and departure time coordination. arXiv preprint arXiv:1701.01391.
5. Van De Hoef, S., Johansson, K. H., & Dimarogonas, D. V. (2016). Computing feasible vehicle platooning opportunities for transport assignments. IFAC-PapersOnLine 49.3, pp. 43-48.
6. Su, D., & Ahn, S. (2017). In-vehicle sensor-assisted platoon formation by utilizing vehicular communications. International Journal of Distributed Sensor Networks 13.7.
7. Thomas, R. W., & Vidal, J. M. (2017). Assessing the Credibility of Vehicle Data Reported by Anonymous Sources. IEEE Ubiquitous Computing, Electronics & Mobile Communication Conference (pp. 613-617). IEEE.
8. Contributors, W. (2018, 10 20). Wikipedia. Retrieved from Prisoner's Dilemma: https://en.wikipedia.org/w/index.php?title=Prisoner%27s_dilemma&oldid=864965858
9. Knight, V. (2018, April 29). Axelrod-Python. Retrieved from Axelrod: https://github.com/Axelrod-Python/Axelrod
10. Axelrod, R. (1984). The Evolution of Cooperation. New York: Basic Books, Inc., Publishers.
11. Contributors, W. (2018, 12 01). Wikipedia. Retrieved from Tit for tat: https://en.wikipedia.org/w/index.php?title=Tit_for_tat&oldid=865044784
12. Lieberman, E., Hauert, C., & Nowak, M. A. (2005). Evolutionary dynamics on graphs. Nature 433.7023, 312.
13. Python Software Foundation. (2012, 04 09). Python.org. Retrieved 12 01,2018, from itertools: https://docs.python.org/3.1/library/itertools.html
14. Russell, Stuart J., and Peter Norvig. Artificial Intelligence: A Modern Approach, 2nd ed. Upper Saddle River, New Jersey; Pearson Education, Inc., 2003, pp.632-633.