Formal Methods for an Iterated Volunteer's Dilemma

Jacob Dineen, A S M Ahsan-Ul Haque, Matthew Bielskas
Department of Computer Science, University of Virginia, Charlottesville, VA 22904
[jd5ed, ah3wj, mb6xn]@virginia.edu

***Abstract*—Game theory provides a paradigm through which we can study the evolving communication and phenomena that occur via rational agent interaction [1]. In this work, we design a model framework and explore The Volunteer's Dilemma with the goals of 1) modeling it as a stochastic concurrent multiplayer game, 2) constructing properties to verify model correctness and reachability, 3) constructing strategy synthesis graphs to understand how the game is iteratively stepped through most optimally and 4)analyzing a series of parameters to understand correlations with expected local and global rewards over a finite time horizon.**

I. Introduction

We are interested in expressing Volunteer's Dilemma games through Prism Model Checker. This is useful because with this software, one can easily tune game parameters to get intuition of game dynamics. This can allow us to see what setting changes correlate with change in expected reward for each player. Additionally, PRISM can provide us a probabilistic graph that reflects a strategy that is optimal (or approximately optimal).

Previous work [1] modeled a public goods game as a concurrent stochastic game, evaluating optimal strategies under a fixed set of parameters deciding the length of the game and the scaling factor associated with resource distribution. Our proposed model would be similar in that it would be finite state, i.e. each agent can choose to share a discrete portion of their initial resources, but would differ in the fact that the Volunteer's Dilemma is a collective good game. The Public Good Game appears to search for localized reward maximization without explicitly being combative or zero-sum.

The novelty is that, to the best of our ability, PRISM has not been used to study Volunteer's Dilemma in the form of an iterated game. By iterated game, we mean that the game repeats so that the environment experiences a soft reset after each round. The initial goal is to check the correctness of our game implementation, so we can guarantee that a win condition is always achievable. Another interest is to tune parameters and plot expected reward as explained above. And finally we want to see how iterations of the game are reflected in the synthesized graphs. Time permitting, we hope this analysis will guide us toward new questions and experiments that reflect subtleties of the Volunteer's Dilemma game.

II. Background

A. Game Theoretic Scenarios

One-shot games, i.e. Prisoner's Dilemma, can typically be modeled with a simple payoff matrix. Players in the game choose a strategy and act concurrently and independently of one another. Extensive form games model game theoretic scenarios with sequential mechanisms, in which a subsequent player acts once their predecessor makes known their strategy and state transition. Iterated games, or repeated games, are examples of extensive form games and study longer (possibly infinite) time horizons. Both methods have gleaned valuable insight into behavioral economics and rational choice theory, and fuse many respective fields. Stochastic games are argued to be the most reflective of real-world systems, as they are governed by probabilistic dynamics that many situations incur. These games are typically modeled as being extensive form, and arguably produce more interesting results of long-run behavior. These dynamics have been studied in games involving social welfare (public goods), robot coordination and investing/auction scenarios [2]–[5].

B. The Volunteer's Dilemma

In game theory, the volunteer's dilemma is a game played by multiple agents concurrently that models a situation in which each agent has one of two options:

[1] prismmodelchecker.org/casestudies/public_good_game.php

1) Cooperation: An agent can make a small sacrifice for public good i.e. that benefits everyone,
2) Defect: An agent can wait and freeride, and hope someone else will eventually cooperate.

The agents make the decisions independently of each other. The incentive for an agent to freeride is greater than the incentive to volunteer. However, if no-one volunteers then everyone loses. Conversely, if at least one person volunteers then everyone receives benefit.

A typical payoff matrix for the volunteer's dilemma looks like this:

TABLE I
PAYOFF MATRIX

	at least one other cooperates	all others defects
cooperate	0	0
defect	1	-10

As stated, the agents have more incentive to defect (1 in this case), than to cooperate (0 payoff). However, if everyone defects, everyone receives -10 payoff.

The volunteer's dilemma occurs in various natural scenarios. For example: in a group of meerkats, some act as sentries to let everyone else know if there are any predators nearby. In doing so, those become more vulnerable. It is also important to understand group behaviors. One particular example we are interested in is a democratic election. Let's assume an election where a candidate has much more supporters than all other candidates. Thus, the supporters of that candidate have little incentive to go out and vote, since that candidate is predicted to win anyway. However, if all of his supporters think in that way and do not vote, that candidate may end up losing the election.

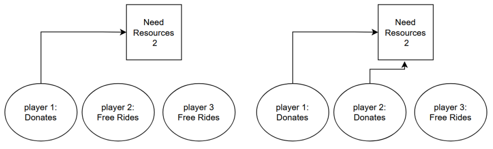


Fig. 1. Volunteer's Dilemma. **Left:** We have a situation where the number of cooperating agents within the system is less than the total required resources for collective group benefit. In this case, no agent in the system benefits. **Right:** Here, the number of agents within the system who choose to cooperate is $\geq$ the total number of resources needed. All agents operating within the system, even those choosing to defect, benefit.

III. Design Overview

Concurrent stochastic multi-player games (CSGs) are an an extension to stochastic games (SGs) popularized in the 1950s. SGs are generalizable to n-player games and present a viable way to model group dynamics, in collaborative or competitive games, where the environment changes given feedback from agents in the system. Beginning from some state $s \in S$, immediate payoff, or reward, is dependent on the actions taken by all agents in the system $v \in V$. Stochastic multi-player games (SMGs) are turn-based and are governed by individual or joint state transitions, where a player chooses from a set of probabilistic transitions to determine the next state [6]. Formally, a CSG can be represented by a tuple not dissimilar from a Markov Decision Process (MDP):

$$\text{G} = (N, S, \vec{S}, A, \Delta, \delta, AP, L)$$

where: N is a finite set of players. S is a finite set of states. A is a finite set of actions available to v_i at time t. Δ is an action assignment function. δ is a probabilistic transition function. AP is a set of atomic propositions, and L is a labeling function. In a CSG, similar to how a policy resolves nondeterminism in an MDP, a strategy resolves choice [3].

Our focus is in games where this transition occurs as a product of all agents simultaneously, hence the concurrence. As the environment changes depending on these actions, the choice of a new state is also influenced, and, in turn, expected future payoffs are affected. The design of our CSG is detailed in this section. We use an extension to Probabilistic Symbolic Model Checker (PRISM), PRISM Games [6]–[8], throughout experimentation.

A. Game Parameters

1) k: We refer to rounds of the finite length game as episodes. $k \in \{1, 2, 3, ..., k_{max}\}$
2) k_{max}: The maximum number of episodes specified as input into the environment.
3) V: The set of agents within a system. The game is typically played with $|V| > 2$ in literature. Here, we fix $|V| = n = 3$. We'll be studying this problem through the lens of a 3-player game.
4) r_{init}: The initial allocation of a resource. Each agent within the system is initialized with r_{init} = 100 at each episode. Resources could be generalized to be currency, votes or public goods, etc.

5) c_i: The current resource allocation for agent v_i at round k. c_i is updated throughout gameplay.
6) s_i: The number of shared resources for agent v_i at round k. A player can donate increments $(\{0, 0.5, 1.00\})$ of their procured resource allocation. $s_i \leq c_i$
7) r_{needed}: A specified parameter that dictates the number of resources needed to 'win' a round k in the game. In the traditional game, only a single volunteer is needed. Here, we consider the effects of resource procurement over finite-length runs of the game. E.g., rewards distributed at round k can be used as 'donation' at round $k+1$. In literature, to reach a winning condition, we generally require donations from strictly less than the total number of agents in the system. This holds here. We require $r_{needed} < 100n$. This parameter is fixed round over round, i.e., it is not dynamically dependent on the values of state variables c_i.

B. Action Space

We discretize the allowable actions, i.e. resource donations, to reduce the search space. We present the action space $\{a_0, a_{50}, a_{100}\} \in A$ below.

TABLE II
VOLUNTEER'S DILEMMA ACTION SPACE

variable	action	definition
a0	Free Ride	A player here chooses to contribute nothing to the pot of r_{needed}. This player is known in literature as a free-rider. They are hopeful that total group contribution still results in immediate payoff without sacrificing any of their resource allocation.
a50	Partial Contribution	A player taking this action will contribute $\lfloor (0.5 * c_i) \rfloor$ resources.
a100	Total Contribution	This action entails contribution in totality. All available resources will be pushed toward r_{needed}. An agent taking this action could be seen as altruistic, as they may perceive the good of the many to outweigh the good of themselves.

C. Reward Structure

We present a simple reward structure as follows. At the kth round, all agents starting in s_0 are to concurrently choose an action. For a winning condition to be met, the sum of total contributions from all agents $\Sigma_{i=1}^{n=|V|} s_i$ must meet or exceed the predefined threshold r_{needed} (figure 2).

The immediate reward passed back to each agent subject to a winning condition can be formulated as:

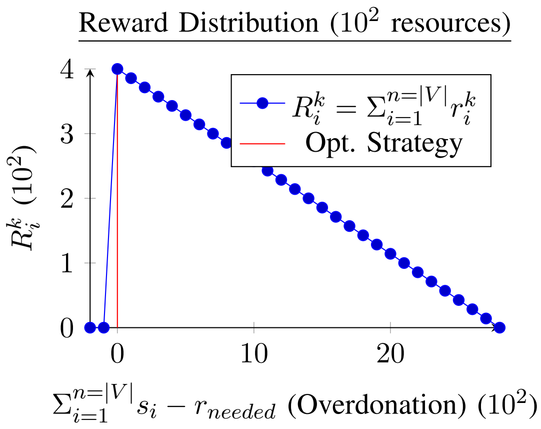


Fig. 2. Reward function given: $r_{needed} = 200, n = |V|, f = 2$. This plot shows donated resources exceeded resources needed and reward (resources) in the 100s of units. When $\Sigma_{i=1}^{n=|V|} s_{i'} < r_{needed}$, that round incurs no reward. When $\Sigma_{i=1}^{n=|V|} s_{i'} = r_{needed}$, an optimal joint strategy has been found. Because a single agent freerode in this instance, the number of resources at the end of this round exceeds those of when the round began. When $\Sigma_{i=1}^{n=|V|} s_{i'} > r_{needed}$, a winning condition has been met, but resources were expended that didn't need to be. This figure shows the linearly decaying reward function, and current resources at the kth round are found via the update function in equation (3).

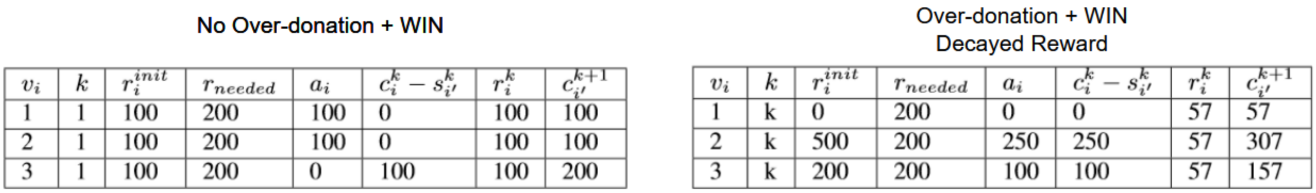
$$r_i^k = \begin{cases} 0 & \left(\Sigma_{i=1}^n s_{i'}^k\right) < r_{needed} \\ \frac{r_{needed} * f}{|V|} & \left(\Sigma_{i=1}^n s_{i'}^k\right) = r_{needed} \\ \frac{(-.014)(\Sigma_{i=1}^n s_{i'}^k - r_{needed}) + r_{needed} * f}{|V|} & \left(\Sigma_{i=1}^n s_{i'}^k\right) > r_{needed} \end{cases} \tag{1}$$

$$R_i^k = \Sigma_{i=1}^{n=|V|} r_i^k \tag{2}$$

$$c_i^{k+1} \leftarrow min(rmax, \left(\left\lfloor \left(c_i^k - s_{i'}^k \right) + \frac{R_i^k}{|V|} \right\rfloor \right)) \tag{3}$$

where the possible resources procured by a single player are constrained by r_{max}. $c_i - s_{i'}$ is the cost incurred by donating resources (initial resources at the start of a round less the donated resources during a round). This can also be thought of as an expenditure. f is a scaling factor to ensure that players who donate aren't penalized more than players that do not donate when a WIN is achieved. $s_{i'}$ is the state that v_i transitions to $c^{k+1} = (s_i, c_i | a_i)$ given their initial round state and the chosen action. Rewards gained at a time-step are re-aggregated in c^{k+1} and are allowable donations in round $k+1$.

In literature, particularly studies involving human psychology, confounding effects may diminish the



No Over-donation + WIN

v_i	k	r_i^{init}	r_{needed}	a_i	$c_i^k - s_{i'}^k$	r_i^k	$c_{i'}^{k+1}$
1	1	100	200	100	0	100	100
2	1	100	200	100	0	100	100
3	1	100	200	0	100	100	200

Over-donation + WIN
Decayed Reward

v_i	k	r_i^{init}	r_{needed}	a_i	$c_i^k - s_{i'}^k$	r_i^k	$c_{i'}^{k+1}$
1	k	0	200	0	0	57	57
2	k	500	200	250	250	57	307
3	k	200	200	100	100	57	157

Fig. 3. Mock Gameplay: Consider a simple model with three players. **Left:** shows the initial run through the CSG. The players transition through the system perfectly and gain max possible global and local rewards. The total resources after a WIN condition are perfectly met are greater than when the round started. **Right:** At the kth round, players have gained resources beyond their initial allocation. A player can donate ($p_{a_i} * c_i$) resources at this step. Players 2 and 3 donate half their resources. They have over-donated, and the reward passed back is less than optimal.

virtue of altruism in and of itself, as it could be done for ulterior motives [9]. We consider this here by 'punishing' over-donations. If $\Sigma_{i=1}^{n=|V|} s_i > r_{needed}$ the immediate reward for all agents at the kth round R_k decays linearly according to the piecewise function noted above. This can be seen in figure 2.

Static games assume a total reset of the environment at each round of an episode. Here, resources gained at round k can be used at round $k+1$. Meaning, the choice to donate is not a binary flag as is usual in models of the Volunteer's Dilemma. We propose that such modifications will allow for more dense analysis of strategy graphs given pre-tuned parameters, as long-run reward aggregation and strategies can be searched for that may shed light on aspects of behavioral theory under specific conditions. Here, we extend a static one-shot version of the Volunteer's Dilemma to an iterated game with the goal of evaluating long-run dynamics (figure 1).

IV. Experiments and Results

The size of the state space is generally represented by $|S| = n^{|A|}$ for static games.. The winning conditions here are dynamically dependent on the state of the game at a given time-step, There are more possible joint policies that result in WIN/SAT as the game progresses and resources are procured via reward feedback, and, as such, the state space grows exponentially over time. E.g., in this first round of a game, assuming $r_{init} = 1$, there are only $\binom{n}{r_{needed}}$ transitions that induce a 'perfect' WIN, where no decay is met via over-donation. $|S|$ increases as $c_i \to r_{max}$. With a parameter set of $\{kmax = 4, r_{init} = 100, r_{needed} = 200, n = 3\}$ the size of $|S|$ grows according to $|S| = 1.6978e^{3.0479k}$. We'll constrain k_{max} to be ≤ 4, as extrapolating this to 5 and 6 rounds leads to $\approx 7mm$ and $\approx 148mm$ possible states.

A. Model Correctness

We look at a mostly fixed parameter set: the number of agents $|V| = 3$, the number of initial resources $e_{init} = 100$, the threshold for resources $r_{max} = 1000$, specified at the local level, and the maximum number of rounds iterated through as $k_{max} = 4$. To ensure that our model is working, we create properties based on a temporal logic, rPATL, which combines PCTL and ATL [10].

With a nonzero probability, we want to ensure that after k rounds, it is *eventually possible* for an agent to have $c_i \geq e_{init}$, which would mean that rewards were accrued during game-play and winning conditions were met. It is not a formal requirement that all agents meet this condition individually, however. If it is not met, the piecewise function in conjunction with the resource update step ensures that $c_i^k < c_i^{k+1}$. If at any point during the game $\Sigma_{i=1}^{n=|V|} c_i < r_{needed}$ it becomes impossible to satisfy this correctness property. Unfortunately, PRISM Games does not support model checking on CTL operators. Ideally, we would want to verify that there exists some state $good = \Sigma_{i=1}^{n=|V|} c_i > r_{needed}$ across k rounds, such that $E[F good]$ evaluates to TRUE. Because this condition is trivially satisfied via the initial state where $n * e_{enit} > r_{needed}$, we look at a case where 2 * r_{needed} are required, which can be satisfied only after the first round of the game.

$$good = \Sigma_{i=1}^{n=|V|} c_i > r_{\text{needed}} * 2$$
$$<< p1, p2, p3 >> P >= 1.0 \, [F <= k_{\max} + 1 \text{ ''}good''] \tag{4}$$

In rPATL, the $<< C >>$ operator specifies a coalition of players [10]. Here, we consider a cooperative game, where players are within a singular coalition aimed at maximizing expected reward. The property in

TABLE III
VGD PROBABILISTIC REACHABILITY ANALYSIS

Round	States	Y	N	M	Y / (Y + N)
1	2 (1 init)	0	2	0	0%
2	55 (1 init)	6	48	1	11.1%
3	1162 (1 init)	141	1009	12	12.3%
4	27065 (1 init)	2724	8766	85	23.7%

equation (4) asserts that there exists a joint strategy, or a collection of policies for each agent, such that the probability of reaching the goal state ”good” within k_{max} steps is at least 1.00. This verifies to FALSE in the first round, and TRUE thereafter to $k_{max} = 4$, suggesting a viable model for our purposes. We can also observe the probabilistic reachability via the PRISM Games GUI for the noted property, detailed in Table III. Intuitively, as the game progresses on, assuming round-wise SAT of the given property, the number of possible states which result in SAT increase. This is due to more resources being injected into the environment, resulting in more possible combinations of donations which result in reward.

B. Property Verification

Now that we're sure our model is implemented correctly in PRISM, the next step is to construct properties and verify them so we formulate a reachability analysis for the CSG. Recall ”probabilistic reachability” as referred to in the previous subsection and Table 3. We note that such information (Y,N,M)/(Yes,No,Maybe) is not directly returned when verifying a property but instead is recorded in the PRISM log. For a probability-based property, the direct result is a Boolean that indicates whether the property holds for at least one state in the model. This corresponds to at least one Yes in the aforementioned (Y,N,M) tuple and we emphasize that both outputs are situationally useful for understanding the game. For a property defined by maximizing or minimizing a variable/reward, the direct result is the max/min number while (Y,N,M) has no reason to be recorded. Below we present some property templates we experimented with in PRISM.

$$<< p1, p2, p3 >> R\{''r1''\}max =?\{F\ k = k_{max} + 1\} \tag{5}$$

With our three players in the game, this property returns maximum reward value $r1$ assigned to Player 1 when the game ends *after* k_{max} rounds. Here $r1$ and $done1$ are interchangeable but $r1$ is able to be examined for all $k = 1, ..., k_{max}$.

$$\begin{gathered} << p1 : p2, p3 >> max =?(R''done1''[Fk = k_{max} + 1] \\ + R''done23''[Fk = k_{max} + 1]) \end{gathered} \tag{6}$$

For this property we have Player 1 aligned against Players 23 for a total of two coalitions. With $done23 = c2+c3-2*e_{init}$, the returned value is the maximum when these two coalitions are separately trying to maximize reward.

$$<< p1, p2, p3 >> P^{\geq 1}[F\ c1 + c2 + c3 < 200] \tag{7}$$

Here we present the first probability-based property. The direct result obtained is 1 if there must always exist a reachable state where the sum of player resources is below 200. In the PRISM log we can examine (Y,N,M) to see the fraction of states where this inequality holds.

$$<< p1, p2, p3 >> Pmax =?[F <= k_{max} + 1 c1 < c2] \tag{8}$$

This property returns the maximum probability that player 2 has more resources than player 1 after k_{max} rounds. This is expected to be 1 since our CSG doesn't impose limitations on how player resources compare *to each other*. Similarly we expect the minimum probability to be zero, and we can obtain a fraction of states that satisfy this from the PRISM log.

C. Reward Maximization

We subject the environment to a property involving global reward maximization: $<< p1, p2, p3 >> R''done123''max =?[Fk = kmax + 1]$, where the label $R''done123''$ specifies the total resources accrued after round k by all agents in the system. The results can be seen in figure 4. Interestingly, we see that when players within the system are instantiated with a lesser initial resource allocation, the maximum possible reward at the end of round 4 is greater than other cases. We believe this relationship to be paradoxical. We can view the slope of the reward plots as an indicator, where $r_{init} < 200$ produces greater rates of change and lesser stabilization as the rounds progress. Because the update step of current resource allocation takes into account expenditures, this leads us to believe that freeriding is a more popular choice of action when initial resources are more scarce. We also note that an optimal strategy is impossible to reach round over round, as the following shows that aggregate reward falls below the ceiling of possible reward $300n$. This is further detailed in figure 5, as we show the optimality of strategies to produce non-intuitive results across 2 rounds of gameplay with

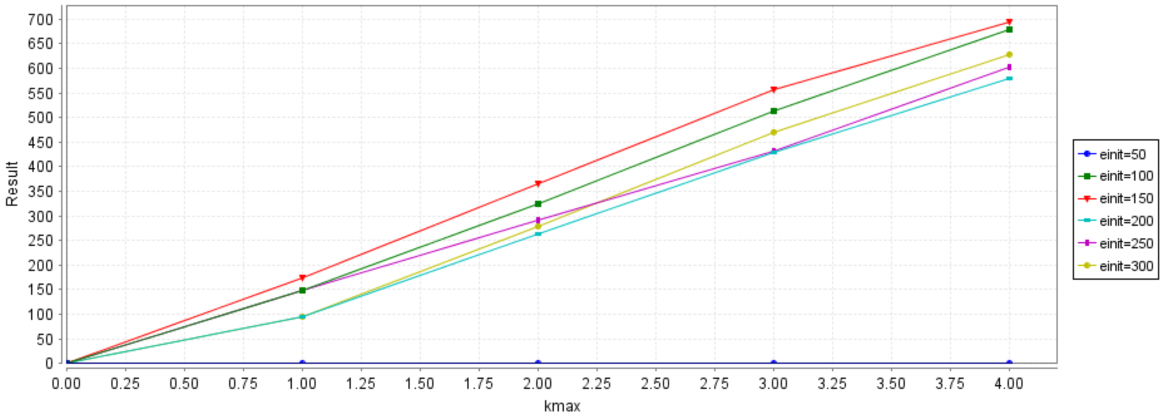


Fig. 4. An iterated run through the system with variable initial resources. The y-axis represents the total, aggregate group reward through time k. The different lines represent varying initial state conditions.

$r_{needed} = 200$. We theorize that group optimality is achieved iff all agents within the system contribute.

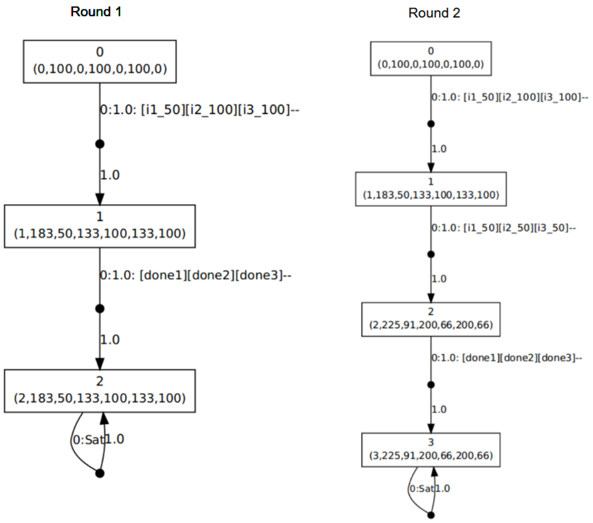


Fig. 5. Strategy Graphs can be used to find an optimal controller given a property. Here, we consider $<< p1 : p2, p3 >> R"r1"max =? [F k = kmax + 1]$ under the specified parameter set noted above. The graphs can be read via [k, c1,s1,c2,s2,c3,s3], where branching is determined by the actions taken concurrently by all agents in the system. Some interesting patterns emerge when looking at global reward maximization against optimal strategies. On the left, results are shown for a single round. From the init state of the game, the optimal strategy is for two players to donate in totality, and for one player to partially donate. On the right, we extend this to round 2. Here, global reward maximization is achieved as a result of full participation via partial contribution. In both cases, no agent within the system freerides.

V. Limitations and Future Work

Reward Properties: Although it's simple enough to formulate properties involving a max or min over linear combinations of rewards, PRISM doesn't support the usage of probability bounds (max/min) or inequalities ($P >= p$) for such formulas. Luckily in this game all rewards are of the form $c_i - e_{init}$ with each c_i a player resource variable. Therefore this became a non-issue as we realized all reward formulas can be substituted if necessary.

Limitations in Multi-Partition Property Analysis: We note that PRISM's support for CSGs is in beta-testing, and additionally the final release may feature limitations to prevent 'obvious' computational intractability. With that in mind, a challenge we faced was the inability to create more than two partitions for properties i.e maximizing sum of player reward. Of course we are still allowed to feature more than two players in a property. But ultimately we lack the capability to fully analyze this game when each player is in a different coalition, and thus we work around this by extracting as much as we can from 1/2-coalition properties.

Strategy Graphs: Perhaps the most interesting analysis involves the strategy graphs generated for specific properties. Because our state space grows exponentially due to the mechanisms involving gameplay, this is exceedingly difficult. For instance, we can look at a strategy graph for one round over three players and the strategy synthesis is easy to conceptualize (reference figure 5). As the game progresses, it becomes computational taxing to conduct value iteration with an exploding state space.

Extensions to Cyberphysical Systems: While we note a theoretical framework above, there are some obvious ties related to free market and democratic systems. For future work, we see an extension of the Volunteer's Dilemma Game (VDG) to the domain of Cyberphysical Systems in proximity-based applications focused on optimal route planning and traffic de-congestion, like Google's Waze [11], [12]. This would present a case of evolving geo-proximity based behaviors and dynamics, as the backbone of these frameworks is dependent on 'guinea-pigs' who willingly and interactively share real-time traffic data. We see these operators as being analogous with cooperators in the VDG, and those who utilize this data to avoid adverse traffic conditions as the defectors. In these cases, we speculate that users who may unknowingly enter situations where the outcome could result in diminished rewards (time, mostly) gain some intrinsic value from sharing valuable information with other users, while the defectors gain reward by utilizing this information to benefit themselves in. Interestingly, such a paradigm presents a case where users are simultaneously collecting information from other users, and also sharing their own information.

VI. Conclusion

We have presented a viable, working model for studying optimal and sub-optimal behaviors in multi-agent systems under probabilistic dynamics. We have also introduced and verified properties to check the correctness of our holistic approach, as well as having analyzed our reward mechanism under various conditions. Our analysis focused mainly on a single parameter set where a number of variables were fixed. The exponential growth of the state space made it difficult to directly induce collaborative strategies that maximize, or minimize, long-run rewards. Our model was presented as a concurrent stochastic game in which players were guided to cooperate with one another. We believe this approach to be realistic, but would like to dive into literature regarding the freerider problem. It is entirely possible that agents in a real-world system do not act cooperatively in the presence of such a dilemma. Perhaps, in the case of a democratic voting schema, a coalition of agents gains intrinsic satisfaction from minimizing the collective reward of an opposing coalition. This could be introduced by partitioning coalitions in the form of subgraphs in a graphical dynamic system. There, it would be of interest to explore such games under a combative approach where coalitions would oppose one another.

References

[1] M. Ummels, "Stochastic multiplayer games: theory and algorithms," Ph.D. dissertation, RWTH Aachen, Germany, Jan. 2010.

[2] O. P. Hauser, C. Hilbe, K. Chatterjee, and M. A. Nowak, "Social dilemmas among unequals," *Nature*, vol. 572, no. 7770, pp. 524–527, 2019.

[3] M. Kwiatkowska, G. Norman, D. Parker, and G. Santos, "Equilibria-based probabilistic model checking for concurrent stochastic games," in *International Symposium on Formal Methods*. Springer, 2019, pp. 298–315.

[4] G. Santos, "Equilibria-based probabilistic model checking for concurrent stochastic games," in *Formal Methods–The Next 30 Years: Third World Congress, FM 2019, Porto, Portugal, October 7–11, 2019, Proceedings*, vol. 11800. Springer Nature, 2019, p. 298.

[5] P. Biró and G. Norman, "Analysis of stochastic matching markets," *International Journal of Game Theory*, vol. 42, no. 4, pp. 1021–1040, 2013.

[6] T. Chen, V. Forejt, M. Kwiatkowska, D. Parker, and A. Simaitis, "PRISM-games: A Model Checker for Stochastic Multi-Player Games," in *Tools and Algorithms for the Construction and Analysis of Systems*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, N. Piterman, and S. A. Smolka, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, vol. 7795, pp. 185–191, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/978-3-642-36742-7_13

[7] M. Kwiatkowska, D. Parker, and C. Wiltsche, "Prism-games 2.0: A tool for multi-objective strategy synthesis for stochastic games," in *Proc. 22nd International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'16)*, ser. LNCS, M. Chechik and J.-F. Raskin, Eds., vol. 9636. Springer, 2016.

[8] T. Chen, V. Forejt, M. Kwiatkowska, A. Simaitis, A. Trivedi, and M. Ummels, "Playing stochastic games precisely," in *23rd International Conference on Concurrency Theory (CONCUR'12)*, ser. LNCS, vol. 7454. Springer, 2012, pp. 348–363.

[9] N. Askitas, "Selfish altruism, fierce cooperation and the predator," *Journal of Biological Dynamics*, vol. 12, no. 1, pp. 471–485, 2018, pMID: 29774800. [Online]. Available: https://doi.org/10.1080/17513758.2018.1473645

[10] M. Kwiatkowska, D. Parker, and C. Wiltsche, "Prism-games: verification and strategy synthesis for stochastic multi-player games with multiple objectives," *International Journal on Software Tools for Technology Transfer*, vol. 20, pp. 1–16, 11 2017.

[11] N. Noerkaisar, B. Suharjo, and L. N. Yuliati, "The adoption stages of mobile navigation technology waze app as jakarta traffic jam solution," *Independent Journal of Management & Production*, vol. 7, no. 3, pp. 914–925, 2016.

[12] S. S. Vasserman, M. Feldman, and A. Hassidim, "Implementing the wisdom of waze," in *International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 24, Buenos Aires, Argentina, 2015 2015. [Online]. Available: http://www.aaai.org/Press/Proceedings/ijcai15.php