Designing Considerate Swarms

Nishant Ramakuru[1] and Namid R. Stillman[1]

[1]University of Bristol, BS8 1TL
ob19208@bristol.ac.uk

Abstract

In the past few years, there has been an increased interest in designing robotic systems that are both fault-tolerant and flexible. However, many applications for these multi-agent systems are highly unpredictable due to the interaction between individual autonomous agents. Swarms of autonomous agents acting towards a common purpose have been shown to perform well when the individual agents are pre-designed to exhibit elements of cooperation and collaboration. Yet many existing models do not explicitly consider how these swarms will coordinate with each other or with other agents from outside the swarm.

In this paper, we adopt a game-theoretic approach to design robot swarms that explicitly considers the behaviour and preferences of other agents on their desired actions. We show that, by including conditionalized actions, agents actively avoid inconveniencing others. In order to demonstrate our considerate swarms, we run simulations of various evacuation scenarios. We find that our swarm outperforms those that use conventional game-theoretic models, evacuating the rooms both faster and more efficiently, whilst also displaying interesting social behaviours.

Introduction

In the past few years, we have seen a rise in robots in industries. Robotics are now being used in various areas such as manufacturing (Realyvásquez-Vargas et al., 2019), search & rescue missions (Arnold et al., 2018), agriculture (Cheein and Carelli, 2013) and defence purposes. Most of the robots are automatic and autonomous, meaning they operate without human intervention. Despite the significant rise of robotics in some industries, there are still considerable roadblocks to the uptake and application of robotics in others. Many situations in the real world are highly unpredictable and small damages to the robot can result in unexpected performance or large sums in maintenance (Zahugi et al., 2012). These issues promote the realization of systems that are both fault-tolerant and flexible.

Researchers have long been inspired by nature to find elegant solutions to modern problems. One of the many fascinating behaviours observed in nature is swarms, defined by (Blum and Merkle, 2008) as a large group of entities that exhibit collective behaviour that align to accomplish a task. Many researchers have found different ways to understand the functioning of multi-agent systems. These include control theory, artificial intelligence, evolutionary models, and game theory (Chu et al., 2010; Vinyals et al., 2019; Byrski et al., 2015; Hernández et al., 2013).

In this paper, we focus on game theory and its application to swarm robotics. We demonstrate a middle ground that retains the game-theoretic assumption of individual preferences but extends the individual decision making to include the interests of others. This added measure to include and consider the outcome of others in the system can be construed as *considerate behaviour*. To test this idea, we simulate strategic behaviours amongst agents in an evacuation scenario, where the agents will be conditioned on the actions of others, henceforth referred to as 'considerate' behaviour. In order to demonstrate the functioning of a considerate swarm, we run simulations of multiple interacting agents within an evacuation scenario. Evacuation scenarios require that each agent is able to leave a confined space without causing danger or harm to others. This paper is organized as follows; In *Background*, we provide the background for game theory in multi-agent systems and the related works on evacuation-related theories and practices. In *Methodology*, we describe our own evacuation model, which contains an environment, target site finding, game strategy, and conflict solving rules. *Results* discusses and analyzes the results from our simulations before we present our concluding remarks in *Conclusion*.

Background

A recent study on cooperative intelligence by Ross Arnold (Arnold et al., 2018), claims that joint behavioural models that depend on an agents' internal mental phenomena are not objective attributes that can be used to model the strategy, as mental states require a context. However, the intentional states may provide adequate reasoning for the behavioural patterns observed. When some of these relevant mental states have been incorporated into each agents preference model, we can understand how these preferences generate coordinated behaviour.

For multi-agent systems, we believe that shared preferences underpins coordinated behaviour. Some researchers argue that specific features, such as team reasoning, are part of a larger conversation on collective intentionality (Searle, 1990; Searle et al., 1995; Gilbert, 1992). They believe that the study of group intentions involves the notion of *"we intention"* and "joint commitments" and hence cannot be constructed from individual-level states. A study by Bratman introduces a concept of "augmented individualism," and asserts that there is no discontinuity between individual and joint intentionality (Bratman and E, 1999, 2013, 1993). Bratman argued that shared intentions primarily consist of inter-related attitudes of the individuals.

In recent years researchers have begun to explore various ways of using game theory to understand this shared intentionality by using game theory as the mathematical framework to understand interactions and behaviour in multi-agent systems (Bacharach, 2006). Coordination in multi-agent systems using the concepts of game theory dates back to the pioneering work of, for example, (Schelling, 1980; Lewis, 2019; Bicchieri, 1997) and many others. The work by (Schelling, 1980), for example, introduced the notions of "tacit coordination," where the "player's" objective is to make contact with the other player through some imaginative process of introspection, of searching for shared clues. Building on their original work, (Sugden, 1993, 2000, 2003, 2015) and (Bacharach, 1999, 2006) coined the term *team reasoning*; the agents identify as a part of a team and modify their actions to align with that of the group.

Traditional game theory (TGT) analyses interactions between agents whilst making minimal assumptions about their individual preferences. Experimentally, an agent is given a set of actions and preferences and these actions produce payoffs. Patterns of these resulting payoffs are then used to find a strategy that maximizes the output. The models are a representation of the decision-making process, where decisions are based on payoffs and individual rationality (Stirling and Volterra, 2016).

Conditional game theory is an extension of TGT which specifies that an agent enters a game with a predefined set of rules but their preferences are then explicitly transformed onto the group-level (Stirling and Volterra, 2016). The key to this approach is to use Bayesian conditionalization to condition an agent's intentions on others. This departs from TGT as the payoff structure is now designed to be conditional on the players' option. This means that, rather than focusing on the results and events after they have occurred, an ex-ante approach (which looks at future events based on possible predictions) is incorporated into individual models to allow for group-level coordination to emerge endogenously. In this paper, we use conditional game theory to motivate the concept of considerate behaviour. On the one hand, considerate behaviour differs from the concepts of team reasoning as it does not require the agents to undergo agent and utility transformations from individual to the swarm as carried out by, e.g. (Bacharach, 2006). It also differs from the concept of intentional cooperation (where agent strategies are explicitly defined to benefit the group). Additionally, it differs from conditional game theory as it does not utilize Bayesian conditionalization, using a notion of social influence instead.

The need for coordination is known to be crucial in a well functioning swarm. As explained by Wynn Stirling (Stirling and Volterra, 2016), coordination is a principle of behaviour on a par with, but different from, performance. Individuals perform; groups coordinate. The performance deals with operational measures of efficiency and effectiveness of individual behaviour. However, this is critical for analysis. In terms of collective functionality, the desire of a group to coordinate is more relevant than the desire of the individuals to maximise personal gains. This can be explained as, how it is more relevant for the team to win a game than for each player to maximize their individual score. Therefore, having intelligent agents with high individual performance but with no cooperation might lead to a poorly functioning swarm, whereas, poor performance yielding agents with high coordination may emerge as a good functioning swarm. As highlighted in (Weiss, 1999), coordination is one of the critical functionalities needed to implement a multi-agent system, especially when the agents are heterogeneous (diverse behavioural strategies) and autonomous (self-controlling and not governed by outside forces). Coordination can briefly be summarized as an increased focus on group-benefit in the trade-off between maximising self and group-benefits. This highlights the difference between coordination (how well agents work together) and consideration (how agents seek to maximise collective aims). Coordination is something that is evaluated post-performance, for example, the average number of passes between players, while consideration is a property of the individual agents, for example, the number of times an individual player passes in a team. In other words, coordination is needed to achieve egotistic goals, whereas, consideration is a behavioural attribute that levels the importance of the system's goals to its own. We can explain this from a design perspective where consideration is a feature for designing robotic swarms whereas coordination is a metric for evaluating them.

The evacuation problem is one of the situations where the trade-off between considerate and selfish behaviour is crucial. If everyone tried to evacuate simultaneously and tried to reduce their individual evacuation time, this might cause congestion and conflicts, and result in higher evacuation time for the entire group (Helbing et al., 2002). A comparison brought forward by (Zheng et al., 2009) shows us several methodological approaches for crowd evacuation. These approaches include lattice gas models, fluid-dynamic models, social force models, cellular automata models, agent-based models, animal behavioural models, and approaches based on game-theoretic models. This highlights the complexity and richness of modelling crowd evacuation scenarios.

It is clear that the evacuation time of the entire group depends on individual interactions. This can be streamlined by minimizing conflicts, only possible if agents exhibit considerate, rather than selfish, behaviours. Mutual cooperation can reward the cooperators and make them exit the room quicker than the defectors, as highlighted by (Bouzat and Kuperman, 2014). With the help of a few strategies, game theory can mimic the behaviour patterns observed in individuals in a panic situation, where the agent might only consider their benefits and try to get out of the situation as fast as possible. However, agent behaviour strongly depends on other agents' behaviours and actions, as the actions computed are a result of interactions amongst agents (Lo et al., 2006). The interactions eventually lead to conflicts, as each agent will prefer an optimum route to the exit, which does not always lead to an inevitable overlap. Game theory is acknowledged as a tool for interpreting these conflicts and, hence, is also a useful tool for studying crowd evacuations (Zheng et al., 2009).

Here, we combine game theory with a notion of considerate behaviour, which we refer to as considerate game theory (CGT). We achieve this by calculating the social influence of each individual agent and using this to alter the agent's utility, and hence, the payoff matrix. We demonstrate CGT on evacuation scenarios, highlighting how game theory can be used to both resolve conflicts and capture features of team reasoning.

Methodology

Before introducing our model of considerate behaviours, described by conditional game theory, we first explain how we apply game theory in a dynamic and then spatial environment. We explain the iterated prisoner's dilemma (IPD), used to model multiple interactions over time, and the role of various strategies within IPD. We then describe the spatial environment, simulation details used to model a multi-agent evacuation scenario. Next, we move to describe how traditional game theory (TGT) is used within the evacuation simulations, before CGT and how this extends TGT to include considerate behaviours.

Iterated prisoners dilemma

Iterated prisoners dilemma (IPD) is a situation where the players play against each other for multiple instances. This enables the players to be able to learn the opponent based on their previous encounters and introduces a social dimension into agent-agent interactions. This is because each interaction is a way for agents to learn the behaviours of other agents. IPD is an ideal method to study the evolution of co-operation among individuals, where we assume that each individual is attempting to maximise their individual payoff. IPD consists of the application of various strategies over multiple iterations such that two players play against each other for a predefined number of times. Here, each game (ie Prisoners Dilemma) is known as an interaction and each iteration is known as a generation. By considering multiple interactions over time IPD can lead to unusual behaviours. For example, always defect is the optimal strategy within the prisoner's dilemma. However, for IPD, the players learn this strategy and adjust their actions accordingly. There are many strategies developed for IPD. We consider the following:

- *Random*: Randomly choose to either defect or cooperate.
- *Always Defect*: Always defect no matter what the opponent's decision is.
- *Always Cooperate*: Always cooperate no matter what the opponent's decision is.
- *Tit For Tat (TFT)*: The opponents who cheated in the previous generations are stored in a list and the player will defect or cooperate based on the previous decisions of their opponent.
- *Exploit*: The opponent will defect if the opponent is found in the list of gullible players. A player is added to the gullible list when the player cooperates if the opponent defects.
- *Grudger*: The player will remember the opponent's decisions from previous generations and uses this information to "hold a grudge", i.e.e they will defect if the opponent has defected in previous interactions.

The *tit-for-tat* strategy was the most successful in Axelrod's original IPD competitions (Axelrod, 1987) and many of the other strategies were an attempt to beat *TFT*.

Different IPD strategies use different lengths of historic interactions. *TFT*, for example, is a one-memory strategy, only making use of information from the previous stage of interaction in making its next decision. Intuitively, the players with longer memories can perform at least as well as those with shorter memories. However, it is unclear whether longer memory strategies outperform shorter memory strategies. A non-trivial question is whether or not a longer memory grants a strategic advantage in IPD. Experiments have revealed that memory helps learning and cooperation in evolution (Yao and Darwen, 1994). For example, it was shown in (Press and Dyson, 2012) that the shortest memory strategy sets the rule of 2-player IPD games. That is, longer memory strategies do not have an advantage over shorter memory strategies in 2-player IPD. However, this result cannot be extended to N-player, where N > 2, as one-memory strategies are not as strong as the popular IPD strategies. This sheds a light on how the memory of the players plays a role in the game outcomes. We explore the influence of memory on a range of different strategies in Section 4.

In order to implement IPD in our simulations, we first simulate interactions in a spatially independent way. We give each player the following attributes; id (to identify the player), points (the overall summation of the score) and strategy (the strategy the player will follow). The players are initiated in such a way that the strategies are uniformly distributed. A metric to note the frequencies of each strategy is initialized, which is equally zero for all the strategies at the start of the game. In contrast to the study carried out by (Stewart and Plotkin, 2012), each strategy will aim to maximise its payoff without trying to alter strategies of other agent's in the system, as conditioning decisions of other agents will be included in the CGT algorithm externally instead of blending it with the player strategy, as explained later. In each generation, a random number of players are picked and play against each other, representing an interaction. Based on the outcome of each interaction, the points of the players are either increased or reduced. After all the interactions are carried out one generation is completed. Using similar policies as those used in genetic algorithms, the stronger players are replicated and the weaker players removed at the end of every generation. This means that the frequencies of strategies belonging to stronger players increases while frequencies of strategies of weaker players decreases. This is done to ensure that the stronger strategy prevails. When extending to the evacuation scenario, interactions occur based on spatial proximity, as we describe below.

Evacuation

We next describe how we simulate IPD within a spatial domain. To investigate the influence of different strategies, we have developed a simulation environment mimicking an evacuation scenario. We choose to write the simulator in Python and follow object-orientated design principles throughout, where each object (player) has 10 attributes: id, strategy, action, states, next state, current state, points, score, data frame, escaped as explained in Table 1. Agents, with random initial positions, are initialized such that there is an equivalent mixture of all strategies. The simulation is then run for a pre-specified number of generations, T, and where two sets of algorithms (based on TGT and CGT) underpin how strategies are executed, which we describe below.

The evacuation room is created using an array of 20 X 20 cells. Each cell value is either 1 or 0 based on if the cell is an open cell or a block cell. Block cells are part of a wall or an obstacle and cannot be navigated by the agents. The agent can move in any one of the four ordinal directions (up, down, left, and right) and each move is determined by the payoff of each action. The payoffs are calculated using a static field $S(i,j)$ as in (Kirchner and Schadschneider, 2002) and shown

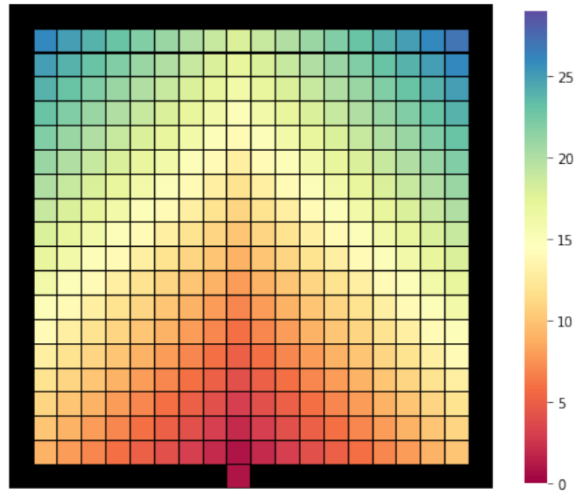


Figure 1: The static field used to calculate payoff for an agent. The cost of the cell is inversely proportional to the payoff of that cell, which shows a higher payoff as agents approach the exit (centre bottom).

in Figure 1, where the static value of each cell is the cost of that cell, which is the inverse of the payoff metric. Simulation parameters, the number of interactions, number of generations, number of players, exit door location, exit door width, number and size of obstacles are varied throughout the simulation to test the robustness of both algorithms. As shown in Figure 1, the payoff for each cell can be calculated according to distance from the exit measured by Manhattan distance metric.

The interaction dynamics are as shown in Figure 2. This figure shows us the outcome of all possible conflicts after one timestep, with red circles representing players with stronger strategies and green circles being players with weaker strategies. When two or more agents attempt to occupy the same position, the agents will be prone to conflict, leading to a one-off game. If an agent is successful then they will move to that cell. All other agents then remain idle for

Id	Represents the Identity number of the agent.
Strategy	The strategy the agent will exhibit.
Action	An array of all previous actions.
States	An array of all previous states.
Next State	Coordinates of the next state to be occupied by the agent.
Current State	Coordinates of the current state of the agent.
Points	An integer that represents the points the current generation.
Score	An array of points scored in each interaction.
Dataframe	Contains all information of the agent (For post-game analysis).
Escaped	Boolean expression to determine the evacuation of the agent.

Table 1: Explains the attributes of each agent.

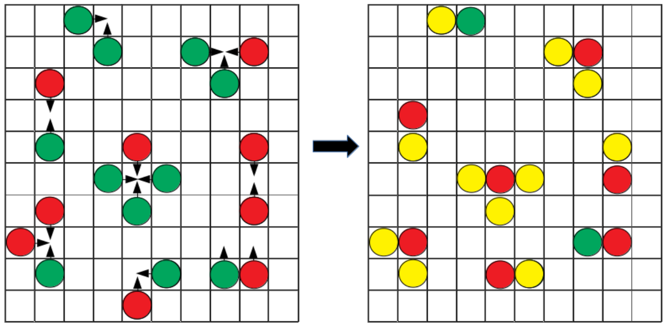


Figure 2: Example of game dynamics after one time step. In this example, agents that are governed by weaker strategies are green, agents governed by stronger strategies are red and idle agents are yellow.

that timestep.

Traditional Game Theory (TGT) Algorithm

Having described both the dynamic and spatial aspects of our simulation, we now introduce the traditional and considerate versions of game theory that are used to describe interactions between agents. As described in subsection Evacuation, each agent, i, has a choice, $S_{i(t)}$, which contains movement in one of four directions, i.e. to go *up, right, left* and *down*, denoted as $\{s^1_{i(t)}, s^2_{i(t)}, s^3_{i(t)}, s^4_{i(t)}\} \in S_{i(t)}$, where i is the agent id (to identify the agent) and t is the time step. Depending on which action gives the maximum payoff P, the agent then moves in that direction. If two or more players are going to occupy the same positions, they then play a game against each other, with the same process as described in *Iterated Prisoners Dilemma*. The outcome of these interactions is shown in Figure 2. The resulting states are based on the outcomes of these games. The combined utility for agent i is given as

$$U(t) = \max(P(S_{j(t)})) \tag{1}$$

where an agent, i, makes a choice $S_{i(t)}$, by maximizing their utility $U(t)$ to occupy a new location based on the payoff (given by the outcome of a game) among j agents within some vicinity. The choice to move can be written explicitly as

$$S_{i(t)} \xrightarrow{U(t)} S_{i(t+1)} | S_{i(t)} = \{s^1_{i(t)}, s^2_{i(t)}, s^3_{i(t)}, s^4_{i(t)}\}, \tag{2}$$

where, for s^n, n represents the number of possible choices (which is up, down, left, right, respectively).

Considerate Game Theory (CGT) Algorithm

The considerate game theory (CGT) algorithm is an extension of the TGT algorithm. When a player in the TGT algorithm makes a decision, they only consider their payoff for one of the four directions. In the case of CGT, there is an additional rule considering the preferences of other agents, i.e. the possible actions of other agents along with the payoff matrix for the said agent are taken into consideration. Before making a decision, each agent, for example, A, measures their utility in the respective directions of all the other agents, as a function of their distance. A then moves in the direction of their highest utility which is the combined probability of other agents to move in that direction. Based on this predicted movement, all the agents have one most probable next state, and A makes a decision considering both this and their own payoff.

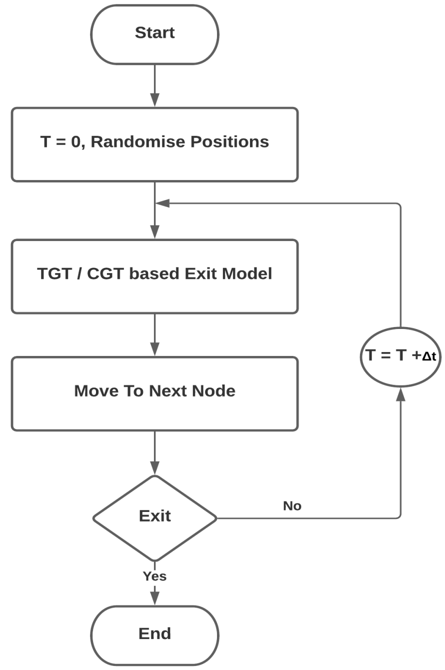


Figure 3: Procedural execution of the algorithm

In order to explicitly calculate the social influence from other agents, we use a relation value $\lambda(i, j)$ for two agents i, j and where $i \neq j$. The relation value is dependent on the vicinity of other agents. For example, if agent A is close to agent B, then the relation value for both of these agents will be high, and their possible actions will play a crucial role in influencing each other's movement. If an agent C is far away from both A and B, then C will play an insignificant role in A and B's decision. We can write this explicitly as

$$\lambda(i, j) = \frac{1}{(s_i(x) - s_j(x))^2 + (s_i(y) - s_j(y))^2} \tag{3}$$

where, $\lambda(i, j)$ is used to calculate the relation value between agents i and j. We use this relation value to modify the original utility of the agent according to TGT, given in Equation 1, by incorporating the probability of conflict Φ, as below

$$\Phi(i) = \sum_{n=1}^{N} \max(P^{'}(S^n_{i(t+1)})) \cdot \max(P^{'}(S^n_{j(t)})), \quad (4)$$

where, $\Phi(i)$ is defined as the product of weighted pay-offs $P^{'}$ for all n cells adjacent to agent x_i (i.e. the possible pay-offs in the next time step) and the weighted pay-offs of all agents that can occupy the said cell. The weighted payoff for one of the actions for agent i, for example, agent i moves one cell up (S^1_i), is then given as

$$P^{'}(S^1_i)) = \frac{P(s^1_i)}{P(s^1_i) + P(s^2_i) + P(s^3_i) + P(s^4_i)}. \quad (5)$$

Using Equations 3 and 4, we can now calculate the utility, $\upsilon_i(t)$, according to the CGT algorithm. We have

$$\upsilon_i(t) = u_{i(t)} - \sum_{j=1}^{\Gamma} \Phi(x) \cdot \lambda(i, j), \quad (6)$$

where Γ is all agents within the system and where $i \neq j$. This is the modified version of the utility given by the TGT algorithm, as described in Equation 1. Comparing Equations Equation 1 and Equation 6 we see that the individual utilities of an agent in TGT and CGT differ, as CGT considers the product of the relation value as well as the probability of conflict. This implies that each agent's utility will be based on the actions of other agents in the vicinity, in some cases causing the agent to choose a lower payoff in order to increase the overall pay-off of the group.

N-Player Games

Finally, both the algorithms are tested in an N-agent scenario, where more than two players can interact. As we consider only ordinal directions, N can be at most three and we consider all three-player interactions as a tournament, such that each player will play against the other two and the winner of the tournament will then be allowed to choose first and so on. In both TGT and CGT, the algorithms are initiated by creating 200 agents with an equal randomised distribution of the strategies. Based on our analysis of the 2 player games with memory, a shorter memory gives the largest variety of different strategies and, hence, more adaptability within the population. We could even note that this is expected to increase the robustness of the successful strategies to changes in the environment. Hence agents are given a limited short memory. In each generation, out of the 200 agents, 150 agents are chosen at random and placed at random positions in the evacuation room. The same set of positions are used for TGT and CGT to make the results comparable. The evacuation simulation is then run with these 150 agents and the strategy frequencies are updated after each generation. This is carried out for 100 generations. The simulation procedure, shown in Fig. 3, is as follows:

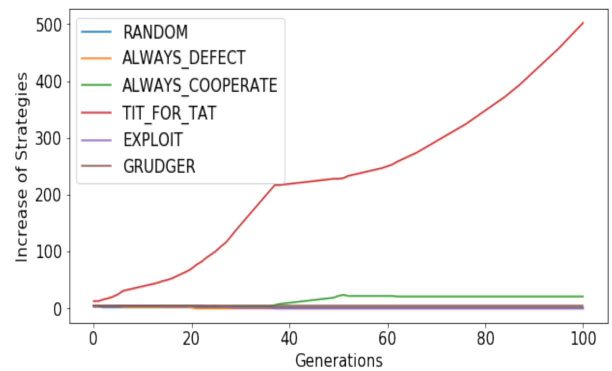


Figure 4: Strategy frequencies

1. Randomise initial positions.
2. Check conflict agents and resolve conflicts by engaging in games based on the governing algorithm (TGT / CGT).
3. After obtaining unique states, move to the next node.
4. If the agent has passed through the exit, terminate the agent, if not, increment time step and go to Step 2.

We characterize the evacuation dynamics of the system by computing the mean exit time, defined as the average number of time steps taken by all agents to evacuate the room. This process is tested over various scenarios, varying the number of agents, the exit door location and the width of the exit and the shape, size and the location of the obstacles, as we describe below.

Results

In order to evaluate the effectiveness of different strategies we first compare strategies in the IPD, without considering spatial effects. As shown in Figure 4, we see that the *TFT* strategy prevails. This agrees with the results observed by Axelrod (Axelrod, 1987). We next consider how 'memory' affects these results, as shown in Figure 5. We observe a drastic increase in the frequencies of *TFT* and *grudger* when memory is restricted. This is because these strategies more frequently punish opponents for choosing to defect (but they do not consider the consequences of these choices on the group). Hence, these strategies tend to win games in subsequent rounds. Always cooperate also increases in frequency as the agents try to help one another.

Few strategies for iterated prisoners dilemma depend on the previous interactions. Strategies like *TFT* only make use of recent interactions, as they mimic the opponent's choice from the previous interaction. However, strategies like the *grudger* depend on a longer period of previous interactions. These properties are vital in an *N*-person game. From Figure 5, we can see that *TFT* prevails, with *grudger* and *always cooperate* both performing well. When the memory length is unlimited, the players with the *grudger* strategy remember the opponent's previous defections and continue to betray each other. This lessens the strength of this strategy. In the case of long memory, the *grudger* strategy is more similar to *TFT* as agents forget older interactions and this allows them to cooperate after some time has passed. The *TFT* and *grudger* strategies are the only strategies with significant frequencies, as *Always cooperate* reduces in frequency. The *grudger* and *TFT* eventually mirror one another as the majority of the players either exhibit *TFT* or *grudger* strategy. Hence, for every interaction, the success of one strategy will cause the other to lose. This results in a zero-sum game.

Finally, when memory is shorter, we see that the *grudger* strategy is able to compete with the *TFT*, as the players forget recent interactions. This makes the player following a *grudger* strategy able to defect only to those who recently betray them and makes it very similar to that of *TFT*. This demonstrates that not having any direct contact with a player may affect strategy success and the actions of players in a group will affect the overall outcome of the group, even when there haven't been any direct interactions between players.

Evacuation Scenario

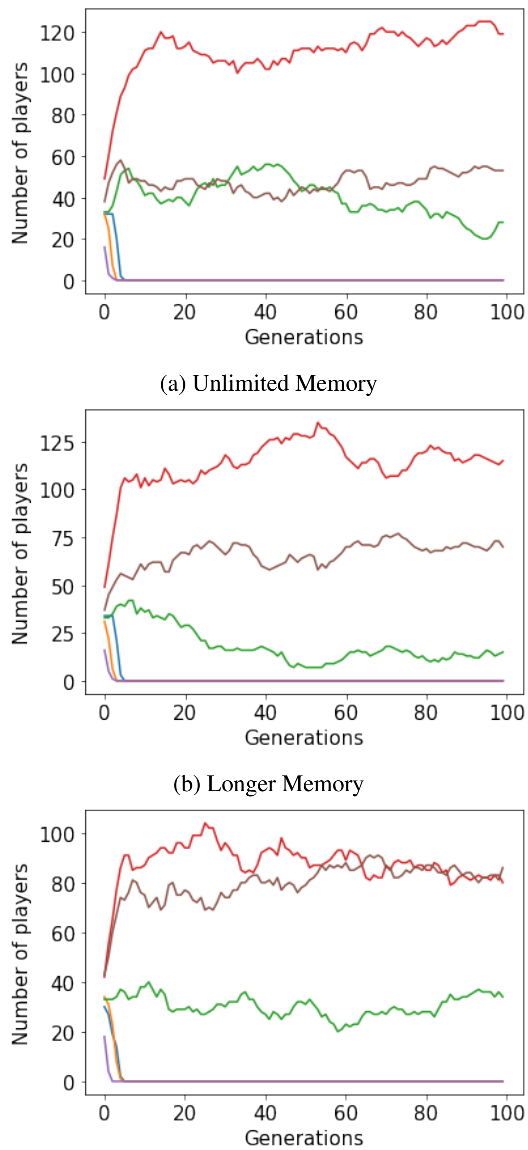
We next discuss how the IPD varies when applied to an evacuation scenario. We start with two-player evacuation, which we then extend into the $N > 2$ games evacuation. The Figure 4 shows the increase in the *TFT* strategy for the player interactions in the 2 player evacuation.

It is interesting to consider the overall behaviour of agents within our simulations. As shown in Figure 6, we observe that, when using the TGT algorithm, agents follow a central line and end up in a congested situation. However, when using the CGT algorithm, agents try to avoid this congestion and instead align themselves with the wall. This increases the distance between the agents and reduces the number of collisions. This result was demonstrated repeatedly and was tested with similar initial conditions. These observations illustrate how the incorporation of considerate behaviours can cause agents to try to avoid conflicts, thereby exhibit cooperation, without being explicitly programmed to do so.

Overall, as shown in Figure 7, we found that 81 times out of 100, the CGT algorithm outperformed the TGT algorithm, evacuating the rooms both faster and more efficiently. The robustness of the algorithm was also tested by changing the evacuation room by adding static obstacles at random positions with random shapes (not shown here), and found that the results matched those observed in Figure 7.

Conclusion

In this paper, we demonstrated that the introduction of considerate behaviours causes agents to actively avoid conflicts. We also demonstrate that the length of memory affects the evolution of strategies, which in turn affect the efficiency of the evacuation of agents. We find that our 'considerate swarm' outperforms those that use traditional game theoretic models, evacuating the rooms both faster and more efficiently. We believe that these findings shed light on how cooperative behaviour can enhance multi-agent systems. Future work will develop the considerate approach to explicitly consider conditionalized preferences, as outlined in conditional game theory, and test these algorithms in real-world situations with both robot and human agents.

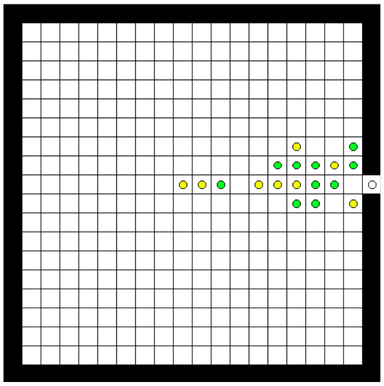


(a) Unlimited Memory

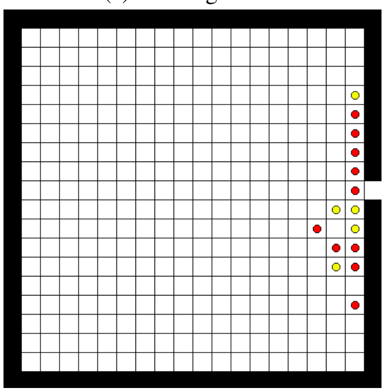
(b) Longer Memory

(c) Shorter Memory

Figure 5: Change in strategy frequencies with respect to number of generations for different number of previous game remembered, including (a) unlimited memory, (b) long memory and (c) short memory.



(a) TGT algorithm



(b) CGT algorithm

Figure 6: Evacuation of TGT and CGT agents.

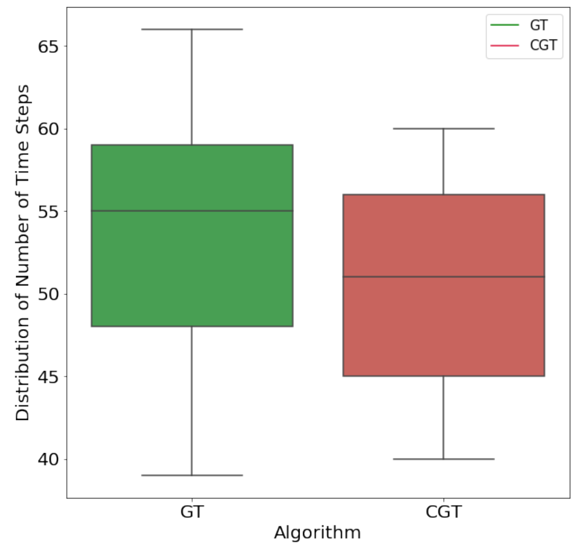


Figure 7: Distribution of number of steps taken by agents with respect to Game Theory and Considerate Game Theory algorithms.

References

Arnold, R. D., Yamaguchi, H., and Tanaka, T. (2018). Search and rescue with autonomous flying robots through behavior-based cooperative intelligence. *Journal of International Humanitarian Action*, 3(1):18.

Axelrod, R. (1987). The evolution of strategies in the iterated prisoner's dilemma. *Genetic algorithms and simulated annealing*, pages 32–41.

Bacharach, M. (1999). Interactive team reasoning: A contribution to the theory of co-operation. *Research in economics*, 53(2):117–147.

Bacharach, M. (2006). *Beyond individual choice: teams and frames in game theory*. Princeton University Press.

Bicchieri, C. (1997). *Rationality and coordination*. CUP Archive.

Blum, C. and Merkle, D. (2008). Swarm intelligence. *Swarm Intelligence in Optimization; Blum, C., Merkle, D., Eds*, pages 43–85.

Bouzat, S. and Kuperman, M. (2014). Game theory in models of pedestrian room evacuation. *Physical Review E*, 89(3):032806.

Bratman and E, M. (1993). Shared intention. *Ethics*, 104(1):97–113.

Bratman and E, M. (1999). *Faces of intention: Selected essays on intention and agency*. Cambridge University Press.

Bratman and E, M. (2013). *Shared agency: A planning theory of acting together*. Oxford University Press.

Byrski, A., Dreżewski, R., Siwik, L., and Kisiel-Dorohinicki, M. (2015). Evolutionary multi-agent systems. *The Knowledge Engineering Review*, 30(2):171–186.

Cheein, F. A. A. and Carelli, R. (2013). Agricultural robotics: Unmanned robotic service units in agricultural tasks. *IEEE industrial electronics magazine*, 7(3):48–58.

Chu, T., Yang, Z., Deng, K., Wang, L., and Xie, G.-m. (2010). Problems in swarm dynamics and coordinated control. *Control Theory & Applications*, 27(1):86–93.

Gilbert, M. (1992). *On social facts*. Princeton University Press.

Helbing, D., Farkas, I. J., Molnar, P., and Vicsek, T. (2002). Simulation of pedestrian crowds in normal and evacuation situations. *Pedestrian and evacuation dynamics*, 21(2):21–58.

Hernández, E., del Cerro, J., and Barrientos, A. (2013). Game theory models for multi-robot patrolling of infrastructures. *International Journal of Advanced Robotic Systems*, 10(3):181.

Kirchner, A. and Schadschneider, A. (2002). Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. *Physica A: statistical mechanics and its applications*, 312(1-2):260–276.

Lewis, D. K. (2019). Convention: A philosophical study—introduction, chapter i, and chapter ii. *Econ Journal Watch*, 16(2):475.

Lo, S. M., Huang, H.-C., Wang, P., and Yuen, K. (2006). A game theory based exit selection model for evacuation. *Fire Safety Journal*, 41(5):364–369.

Press, W. H. and Dyson, F. J. (2012). Iterated prisoner's dilemma contains strategies that dominate any evolutionary opponent. *Proceedings of the National Academy of Sciences*, 109(26):10409–10413.

Realyvásquez-Vargas, A., Arredondo-Soto, K. C., García-Alcaraz, J. L., Márquez-Lobato, B. Y., and Cruz-García, J. (2019). Introduction and configuration of a collaborative robot in an assembly task as a means to decrease occupational risks and increase efficiency in a manufacturing company. *Robotics and Computer-Integrated Manufacturing*, 57:315–328.

Schelling, T. C. (1980). *The strategy of conflict*. Harvard University Press.

Searle, J. R. (1990). Collective intentions and actions. *Intentions in communication*, 401:401.

Searle, J. R., Willis, S., and Marion, S. (1995). *The construction of social reality*. Simon and Schuster.

Stewart, A. J. and Plotkin, J. B. (2012). Extortion and cooperation in the prisoner's dilemma. *Proceedings of the National Academy of Sciences*, 109(26):10134–10135.

Stirling, W. C. and Volterra, V. (2016). Theory of coordinated agency. *Brigham Young University (mimeo)*.

Sugden, R. (1993). Thinking as a team: Towards an explanation of nonselfish behavior. *Social philosophy and policy*, 10(1):69–89.

Sugden, R. (2000). Team preferences. *Economics & Philosophy*, 16(2):175–204.

Sugden, R. (2003). The logic of team reasoning. *Philosophical explorations*, 6(3):165–181.

Sugden, R. (2015). Team reasoning and intentional cooperation for mutual benefit. *Journal of Social Ontology*, 1(1):143–166.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. (2019). Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354.

Weiss, G. (1999). *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT Press.

Yao, X. and Darwen, P. J. (1994). An experimental study of n-person iterated prisoner's dilemma games. *Informatica*, 18(4):435–450.

Zahugi, E. M. H., Shabani, A. M., and Prasad, T. (2012). Libot: Design of a low cost mobile robot for outdoor swarm robotics. In *2012 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 342–347. IEEE.

Zheng, X., Zhong, T., and Liu, M. (2009). Modeling crowd evacuation of a building based on seven methodological approaches. *Building and Environment*, 44(3):437–445.