An Analysis of Social Laws in the Presence of Rogue Agents

Tristram Southey
University of British Columbia
tristram@cs.ubc.ca

Abstract

This paper examines the effects of rogue agents upon multi-agent systems which implement social laws. A taxonomy of rogue agents is suggested and then experimental analysis is made of the effects of 6 types of rogue agents on an existing set of social laws which govern the movement of mobile robots. The implications of the success of each type of agent are examined and various improvements are suggested to both these laws and social laws in general.

Introduction

In 1994 Tennenholtz and Shoham proposed a concept called “social law” for multi-agent societies (Shoham & Tennenholtz, 1994). Social laws are a set of pre-defined rules which all agents in an environment agree to follow which allow them to successfully co-exist. Having such a set of rules is useful as they give guidelines on the behavior of agents while still allowing them a certain amount of flexibility. For example, a group of mobile robots might have a set of rules which define how they should move in order to prevent collisions. If properly constructed, these rules would still allow the robots some freedom as to the exact implementation of their movement algorithms. In addition to this, social laws also allow the agents to have guarantees about worst case and average success. In the mobile robots example, the average and maximum amount of time and distance traveled to reach any particular goal can be determined.

As Tennenholtz and Shoham noted at the end of their paper, one potential problem in the theory of social law is that it relies on every robot always obeying the rules. Rogue agents which violate these laws could potentially destroy many of the advantages that social laws convey. Social laws need to be tested to see how they will perform in the presence of rogue agents.

This paper proposes a set of rogue agents which can be used to test how tolerance of a set of social laws to rogue agents. The Tennenholtz and Shoham 2nd traffic law is used as an example of the analysis process and to describe the general effect of rogue agents on social laws. This paper ends with a brief discussion of different techniques which can be used to deal with rogue agents.

Traffic Laws

Tennenholtz and Shoham created two sets of social laws for controlling the movement of robots on a grid style world (Shoham & Tennenholtz, 1994). Their second set of traffic laws is a good environment for testing the effects of rogue agents as it is well known and analyzed and can be simulated relatively easily. The rule set used here has been slightly modified from that found in their original paper to remove some elements that were unnecessary for our implementation.

The traffic law environment is a grid world of size n x n . The grid is divided into a set of sub grids, each of size *2*m* where *m* is the number of robots and $m = O(\sqrt{n})$. Each robot moves simultaneously. Robots occupy the intersections of the grid and move along the lines. They can move one space each world tick. If two robots enter a square at the same time they have “collided”. The exact effect of collisions is not specified but since the robots should never collide if they follow these rules, this is not an issue at this point.

The social laws are as follows: (see figure 1 for details)

1) On the edges of the sub grids (referred to as the coarse grid), robots move up on even columns, down on odd, left on even rows and right on odd.
2) On the coarse grid, a robot may not make more than k turns, where *k* is some small constant, before it enters a sub grid.
3) At intersections, robots follow a FIFO ordering with precedence given to robots on columns on simultaneous arrivals.
4) Robots enter a sub grid by the lower left corner. They then move across the bottom. If they detect a robot in front of them they wait one tick. When they reach the end of the row they wait *2n-mk* ticks. The robot may then move freely within the sub grid, so long as it reaches the top left in *4m-2* steps and it does not enter the area with the thick lines or leave the sub grid. It then moves along the top row, down the left hand side to the mid point and out.

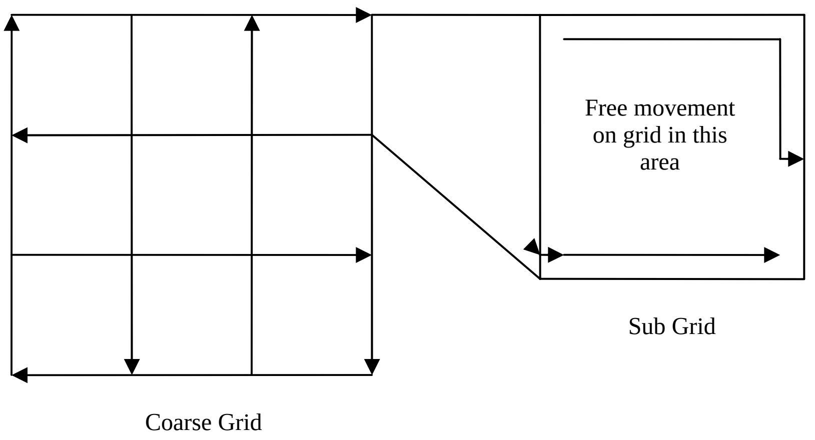

Figure 1: The basic layout of motion in the traffic law. The robots move along the lines on the course gird to reach a sub grid. The sub grid is entered by the bottom left corner. Robots follow that line across and are then allowed to move into the main section. The exit the main part of the sub grid

5) Robots treat the exit point of a sub grid as an intersection.
6) Robots must keep moving unless the location they wish to enter is currently occupied by a robot or unless told to stop by another rule.

These laws give certain advantages to the robots. Most importantly, it guarantees that no collisions will occur between robots. Every coordinate on the grid is reachable (ie, liveness is guaranteed). Within a sub grid a robot is still free to plan its movements to reach multiple goals, so long as it exits within the right amount of time. The average time to reach a coordinate is linear to *n* and there exists a plan which will enable the robot to reach its goal in $t+2n+o(n)$ steps, where *t* is the complexity of the best solution in the absence of other robots. In most cases, the robots will do much better than this and can achieve results close of optimal. [1] However, all the guarantees rely on the fact that all agents follow the laws.

Rogue Agents

Before we look at rogue agents specifically in the context of social law, it will be useful to define a taxonomy for them. For the purposes of this research, rogues agents are defined as any agent that disobeys the social laws for a given environment. We divide rogue agents into three categories based the motivation for their rogue behavior. The three basic classes we define are "Quixotic", "Malicious" and "Greedy":

Quixotic Rogue Agents: These are agents which do not "intentionally" cause harm to the system. Their rogue behavior comes through some sort of error or unusual circumstances. A Quixotic rogue agent which either constantly or periodically violates the rules through its actions called a "Chaotic". A Chaotic's actions need not be random. They could, for example, be following a different set of social laws. An agent which simply ceases to perform any activity at all is called a "Corpse".

Malicious Rogue Agents: There are agents whose purpose is to disrupt the social law society they enter. This may be done in several ways, such as preventing the law-abiding agents already in the environment from performing their tasks or physically harming existing agents or removing required resources. Malicious agents may or may not know what the social laws for an environment are but if they do they may use their knowledge to detect and exploit weaknesses. An agent which seeks to disrupt all agents equally through its actions is called an "Anarchist" while one which targets individuals or a sub set of all agents is called a "Stalker".

Greedy Rogue Agents: These are agents which have some activity they need to perform in the social environment and which may violate the social laws to do so. An agent which does not take into account the adverse impact it has on others in a system is called a "Cutthroat". A Greedy agent that acts to minimize the adverse impact it has on other agents is called a "Gentleman".

These three groups are not exclusive of each other. It is possible for an agent to have a combination of influences. However, for the purpose of testing social laws, the three groups will be considered separately.

It is also important to stress that the rogue agent types defined here do not necessarily represent all possible rogue agents. Rather they are intended to be a useful selection that covers the major types of rogue agents.

Gridworld

Here we use simulation to test the effects of rogue agents on a system. Gridworld was created as a practical implementation of the traffic laws described above and it allows you to simulate the both law abiding robots which obey the social laws and rogue agents. A few key properties of the system are as follows:

Goals: Each robot has a desired location on the grid which they want to reach. Each time they reach one of these points, a new goal in randomly generated on the board.

Collisions: The original work by Tennenholtz and Shoham did not describe the effect on the robots of a collision since their work revolved around the idea that collisions had to be avoided. However, the presence of rogue agents often leads to collisions. In a real world application the effects of collisions could be minor or catastrophic. Rather than have that the simulation end when two robots collide, Gridworld simply allows both robots to temporarily occupy the same square and keeps track of the number of times each robot collides.

Horizon: Each robot has a vision horizon which determines how far they can see. They can detect other robots in a box around them which extends two squares in all directions. Past this point the only thing the robot knows is location of its goal.

Gridworld Rogue Agents

To get a broad spectrum of effects, a rogue agent for each of the sub categories given above was created in the Gridworld environment. This way we can see how all the different natural types of rogue agents will affect the system. Each agent's success can reveal potential failures in the society. For the results of the trials see figure 2.

The Chaotic:
Algorithm: The Chaotic is designed to test a systems tolerance to totally or partially random agents. For the traffic law system we went with a totally random agent as this would indicate how the system deals with the extreme case. Random agents such as these are important to look at as, for most applications, they are one of the most common types. Many multi-agent systems can make guarantees that agents will not enter with either malicious or greedy intent but it is hard to guarantee that no aspect of the programming or physical design of an agent will fail. The algorithm for the robot is simple; it randomly decides to move in one of the four directions or to wait each turn with equal probability and does not worry about collisions.

Results: The Chaotic agents generated some collisions between it and other agents but did not seriously affect either the distance or time to goals. If the cost of collisions is low then the impact of Chaotic agents would be similarly low. However, since the system is primarily designed to minimize collisions at the cost of speed, if the cost of collisions is low, then the social laws would likely be more detrimental than useful.

The Corpse:
Algorithm: Like the Chaotic, the Corpse is another agent which is likely to be found in any system. They test how severe is the average impact of an agent suddenly ceasing to function for some period of time. Since we want to simulate the failure of a normal robot, the Corpse robot follows the traffic laws. However, at each step there is a 1% chance that the robot will stop moving. There is an equal chance each round after that that the robot resumes function. On average the Corpse agent is inactive 50% of the time and stops for around 50 ticks. It is important that the agent move around the grid and follow the rules the rest of the time as this means the distribution of locations it dies at will match the distribution of locations the law abiding agents reach. For the worst case effect of a Corpse robot on the traffic law system, the Anarchist is a better test as it is aimed at the most vulnerable section of the system.

Results: Overall, the impact of the Corpse was not severe. The Corpse robot never generated any collisions, which makes sense given that law abiding robots will never enter a square containing a robot. The only effect the Corpse had was to slightly slow down the average time it took robots to reach their goals.

The Anarchist:
Algorithm: The Anarchist tests how well the system can tolerate an attack aimed at its weakest location. By examination it is clear that the single most disruptive thing a robot can do in the traffic law society is to place itself in the middle of any of the four way intersections and not move from there. The center of the four way intersections has the highest traffic. If a robot blocks an intersection they make it impossible to reach one entire sub grid and more difficult for robots to reach any goal in k steps. Therefore, the algorithm of the Anarchist is simply to move to the nearest four way intersection and stay there for the whole simulation.

Rogue agent	None	Chaotic	Corpse	Anarchist	Stalker	Cutthroat	Gentleman
Goals Reached (mean/worst/best)	M:63402 W:63319 B:63500	M:63329 W:63276 B:63457	M:60712 W: 32217 B:60838	M:1 W:1 B:4	M:63292 W:63205 B:63415	M:63300 W:63247 B:318122	M:63377 W:63274 B:313870
Collisions (mean/worst/best)	M:0 W:0 B:0	M:6022 W:47313 B:5723	M:0 W:0 B:0	M:0 W:0 B:0	M:12063 W:1169090 B:10979	M:2004 W:16152 B:1941	M:0 W:0 B:0
Average Time to Goals (mean/worst/best)	M:157.74 W:157.93 B:157.49	M:157.86 W:158.08 B:157.63	M:164.63 W:310.39 B: 164.37	M:10000000 W:10000000 B:5000000	M:157.70 W:158.05 B:157.42	M:157.93 W:158.11 B:31.43	M:157.78 W:158.40 B:31.86
Average Distance Traveled to Goals (mean/worst/best)	M:129.40 W:129.60 B:129.22	M:129.36 W:129.58 B:129.13	M:129.46 W:129.70 B:129.25	M:144.00 W:250.0 B: 99.00	M:129.46 W:129.77 B:129.18	M:129.48 W:129.64 B:31.43	M:129.40 W:129.62 B:31.85

Figure 2: Trials for each one of the different rogue agents, testing 1 of each type of rogue agent and 8 normal social robots. The grid configuration was set for 8 robots, with sub grids of size 16x16 and an overall grid size of 48x48. Each test ran for 10 million cycles. Chaotic, Stalker and Anarchist agents do not have goals and are not taken into account when calculating mean, worst or best values for goals reached, time to goals or distance to goals. Distance traveled is measured in grid points and time to goals in world cycles.

Results: The effect of the Anarchist on the system was devastating. A few robots managed to reach their initial goals but within 300 ticks every robot had tried to pass through the intersection and was stuck there. The Anarchist will not cause any collisions, apart from those resulting from the Anarchist moving to the intersection. The success of the Anarchist shows that the traffic laws are highly susceptible to general disruption from malicious agents.

The Stalker:
Algorithm: The Stalker tests how well the system can tolerate attack aimed at specific agents. There are two obvious ways that the robot can disrupt a single agent. Either the Stalker can attempt to block an agent from reaching their goals or it can attempt to collide as frequently as possible with the other agent. It is easy to see that in the first case the Stalker can prevent any individual robot from reaching their target either by blocking the entrance to the grid containing it or by placing them self on top of it. Since the Stalker is free to move through the system as it desires, it can almost invariably beat a social robot to its goal. Because of this, the Gridworld Stalkers are of the second type, those that try and collide frequently with other robots. These robots follow the shortest path to reach their target and collide with it. Since collisions only occur if two robots enter the same square at the same time and did not start from the same square, if the Stalker is in the same square as its target, it moves in a random direction out of it and tries to collide again next round.

Results: It is clear that the Stalker agents are able to reach the other social robots and collide frequently with them. The traffic laws are not well designed to prevent malicious attempts at collision. Though it is a somewhat obvious result for the traffic law system it might not be for another set of social laws. The effectiveness of a Stalker type rogue agents needs to be addressed for a system to deal with malicious targeted agents.

The Cutthroat:
Algorithm: In Gridworld, the Cutthroat agent has goals they need to reach like a normal social robot. Their purpose is to minimize the amount of time it takes to reach these goals, so the movement algorithm they follow is always to take the shortest path to their goal, irrespective of social laws and other agents in the way. The success of Cutthroat agents is a good indication of how worth while it is for agents to ignore the social laws.

Results: It is clear that a Cutthroat agent can reach more goals than a law abiding agent. However, the number of collisions that they have is significant enough that it is unlikely to be profitable. This would, of course, depend on the cost of collisions. If an agent were able to act in a Cutthroat manner without any detrimental impact on itself, one might question the need for social laws. Since in this model collision had no cost and that the social robots were rarely blocked by the movements of the Cutthroat agent, there was little effect on the number of goals reached or average time to goal for law-abiding agents.

The Gentleman:
Algorithm: The purpose of the Gentleman agent is to try and reach a series of goals, while causing the minimum amount of interference to the existing robots. Most importantly, the Gentleman tries not to cause collisions. At each step, the Gentleman looks to see if there are any robots within its horizon of 2 squares. If it detects no robots in takes the shortest path to its goal. If it detects robots within its horizon, it constructs an "Intension Map" for the area. In the intension map, each grid coordinate that each other robot within the horizon could be in next

turn is labeled as dangerous. The movements of the other robots are limited by the social laws, so a robot in an even row will move one square right unless it must stop because of the presence of another robot. Once this intension map is produced, the Gentleman determines whether the move which will take it closer to its goal enters a dangerous spot on the intension map. If it does, the Gentleman tries the other directions available. If all directions move it into danger, then the robot will wait at the square it is currently in. There do exist situations where the Gentleman cannot be sure it will avoid a collision by moving, see figure 3.

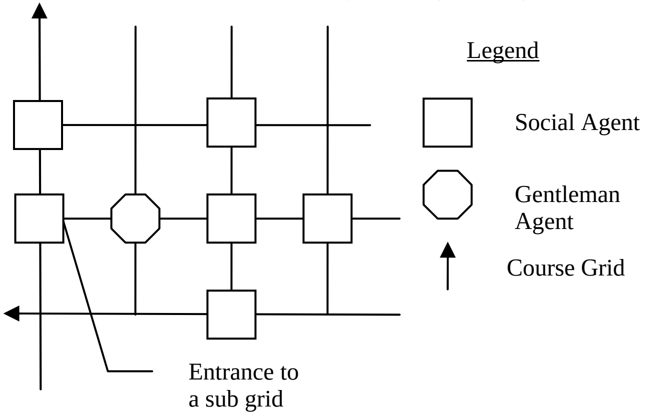

Figure 3: An example of a situation where a Gentleman agent must wait.

Here the Gentleman agent cannot move without the possibly of colliding with one of the other agents. If the agent at the entrance to the sub grid moves right then it will collide with the Gentleman. However, since social agents never enter a square currently occupied by another robot, the Gentleman is always safe if it doesn't move. Eventually the other robots will move away. The possible flaw with this strategy is that the immobile Gentleman might cause a permanent deadlock. In figure 4 you can see that the other three robots cannot move as one would be forced to enter the same square as the Gentleman, one into the same square as another agent and the one exiting the sub grid might collide with the Gentleman. Waiting cannot solve this problem as all the other agents will also keep waiting. However, the Gentleman here knows that the agent at the exit of the sub grid will not move into the square below it and so the Gentleman can move down. There does not exist a situation where the Gentleman can cause a dead lock such that no one can move out of it. This would require each law abiding agent to be prevented moving by the Gentleman and that all the moves possible for the law-abiding agents must pass through the square containing the Gentlemen. There is no location on the board which meets these criteria. Either the social robots will eventually be able to move out or the Gentleman will know that the other agents will not move and it can then move out.

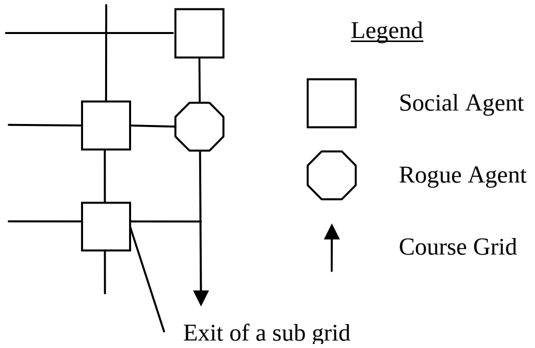

Figure 4: An example of a situation where if a Gentleman waits it will produce a deadlock.

Results: As was to be expected, even with tests of 1 billion cycles, there were no collisions between any of the robots. The Gentleman agent was able to reach nearly 5 times the number of goals any of the other robots reached. This result is particularly problematic as it shows that robots really do have an incentive to violate these social laws as an agent can benefit without suffering any consequences. However, if there are other rogue agents in a society, then problems arise for Gentlemen. The movements of a rogue agent cannot be predicted as easily so they have a wider impact on the intention map. Intension mapping works well because law abiding robots are predictable.

Gridworld Rogue Agent Analysis

Each of the six different rogue agent types showed both the strengths and weaknesses of Gridworld. The system is fairly tolerant to random agents, especially Corpses, mostly because they could not cause collisions. The social laws used are susceptible to malicious attacks either from mobile or immobile agents. Greedy agents that seek to minimize their own impact on other agents can act with impunity but they gain little from being more aggressive and taking the shortest path. On the whole, the system is not tolerant to rogue agents but the simulations and results have revealed some techniques which can increase the systems success.

Rogue Agent Solutions

There exist several solutions for dealing with rogue agents in a social law system.

Flexible Social Laws: Briggs and Cook have adapted the theory of social law and added the concept of flexibility (Briggs & Cook, 1994). In a flexible social law system, each law has attached to it a monotonic importance rating. Agents follow the social laws as normal but when they encounter a situation where they cannot take any action, they discard the law with the lowest ranking until they are able to move.

In theory, this could be applied to the traffic laws. For example, it might be possible to permit robot which cannot move to ignore the rule which says that they must always move in specific directions on the course grid. This would let robots blocked by an Anarchist or a Corpse agent to escape the situation. However, there are several problems with using flexible social laws in this case. First of all, it is quite acceptable for robots not to make a move and the system relies on it. It would be better for the robot to start discarding social laws when they have stopped for *X* turns, where *X* is the maximum number of turns they should ever have to wait if other robots were following the social laws. A tougher problem to solve is the question of how long the laws should be discarded for. If they were only discarded for one turn it wouldn’t always let the robot escape but you don’t want to discard the law permanently. One solution would be to relax the social law for an exponentially growing period each time it encounters the same problem. Finally, there are many problems which arise when you stop following the laws and most guarantees will vanish.

Fundamentally, it is questionable what flexible social laws really give you over regular social laws. They are really just a different way of thinking about and designed social laws and not a particularly useful model for modifying existing social laws.

Cautious flexibility: This is a theory which was derived from the success of the Gentleman agent’s intension map and flexible social laws. When a robot encounters a situation where it cannot move and it has waited longer than the longest possible wait if the social laws were being obeyed, then the robot is allowed to temporarily ignore the social laws. It finds the shortest path which should take it back onto the path it should be following. As it moves along this new path, it uses an intension map like the Gentleman agent to avoid collisions. Once it is back onto its original path, it continues along normally.

The only serious potential problem with this strategy is that other robots might also be currently using the cautious flexibility solution. To deal with this, the robot could either assume that all robots might make any move and build its intension maps appropriately, or it can try and judge whether any particular robot is behaving according to the laws and build probabilistic intention maps. The first technique has a greater probability of having deadlocks and will take longer but should never result in collision. The second will be more flexible but might cause collisions. In the end, cautious flexibility is best at dealing with still obstacles like Corpses and Anarchists.

Central Authority: Both of the techniques suggested so far only really deal with agents that can’t move. What then can be done with moving rogue agents? These are tough as they are highly unpredictable. One possible way of dealing rogue agents is to have a central authority responsible for enforcing social laws and punishing violators. Punishments could take many forms, such as removing the goals for the rogue agent or using the existing law abiding agents to inhibit the rogue agent. If a Greedy agent knew that all the other agents in a system will block its moves if it violates the rules, then it would be less likely to do so. Alternatively, even if the central authority cannot punish rogue agents, it would still be useful for it to alert all the law abiding agents to the location of the rogue agent. This would make cautious flexibility more useful as all violating robot would be marked and they could be circumnavigated more quickly. Also, the intension maps could be more accurate as robots which are not marked need not be considered as possibly moving in every direction. The greatest problem with central authority solution is that it is not always feasible to have a central authority examining the moves of all agents (Murate & Minsky, 2003). Work has been done on the production of decentralized enforcement techniques.

Conclusion

It is possible to analyze the effects of rogue agents on a specific set of social laws through simulation. It is important that a variety of rogue agents be used to test the system as there are so many causes of rogue behavior. The traffic laws examined here were not very resilient against rogue behavior, especially in the form of targeted attacks on weak points or greedy and unobtrusive agents.
Future work might include an expansion and formalization of the rogue agent taxonomy or the construction of more general mathematical ways of determining the effects of rogue agents without the need for simulation.

References

Shoham, Y., and Tennenholtz, M. 1994. On Social Laws for Artificial Agent Societies: Off-Line Design. *Artificial Intelligence.*

Briggs, W., and Cook, D. 1995. Flexible Social Law. *Proceedings of the International Joint Conference on Artificial Intelligence.*

Murate, T., and Minsky, N. 2003. On Monitoring and Steering in large Scale Multi-Agent Systems. *Proceedings of the International Workshop on Large Scale Multi Agent Systems.*