Non-deterministic Social Laws

Michael H. Coen

MIT Artificial Intelligence Lab
545 Technology Square
Cambridge, MA 02139
mhcoen@ai.mit.edu

Abstract

The paper generalizes the notion of a *social law*, the foundation of the theory of artificial social systems developed for coordinating Multi-Agent Systems. In an artificial social system, its constituent agents are given a common social law to obey and are free to act within the confines it legislates, which are carefully designed to avoid inter-agent conflict and deadlock. In this paper, we argue that this framework can be overly restrictive in that social laws indiscriminately apply to all distributions of agent behavior, even when the probability of conflicting conditions arising is acceptably small. We define the notion of a non-deterministic social law applicable to a family of probability distributions that describe the expected behaviors of a system's agents. We demonstrate that taking these distributions into account can lead to the formulation of more efficient social laws and the algorithms that adhere to them. We illustrate our approach with a traffic domain problem and demonstrate its utility through an extensive series of simulations.

Introduction

Agents designed to exist in multi-agent systems in general cannot afford to be oblivious to the presence of other agents in their environment. The very notion of a multi-agent system presupposes that agents, for better or worse, will have some impact on each other. A central problem in Distributed AI (DAI) has been to develop strategies for coordinating the behaviors of these agents – perhaps to cooperatively maximize some measure of the system's global utility or conversely, to insure that non-cooperative agents find some acceptable way to peacefully coexist.

Many coordination strategies have been developed for managing multi-agent systems (MAS). One axis on which we can contrast these different approaches is the degree of agent autonomy they suppose. For example, a centralized planning system [5] might globally synchronize each agent's activities in advance, taking pains to insure that conflict is avoided among them. Agents can then blindly follow these centrally arranged plans without further consideration. Alternatively, at the other end of the spectrum, agents can be wholly autonomous and pursue their individual goals without relying on any centralized control mechanism. In the event conflict arises among them, preformulated rules of encounter allow the agents to dynamically negotiate among themselves to resolve it [3].

An intermediary approach between these extremes has been explored in the development of *artificial social systems* [2,4], whose workings should feel familiar to anyone living in a civilized country. In an artificial social system, its constituent agents are given a common *social law* to obey and are free to act within the confines it legislates. A social law is explicitly designed to prevent conflict and deadlock among the agents; however, for it to be deemed *useful*, it should simultaneously allow each agent to achieve its individual set of goals. Thus, designing a social law is something of a balancing act. It must be sufficiently strict to prevent conflict or deadlock, and simultaneously, it must be sufficiently liberal to allow the agents to efficiently achieve their goals. Useful social laws can be designed that not only avoid inter-agent conflict but also minimize the use of energy, time, and other resources appropriate to the problem domain. Fitoussi has examined an extension of this theory involving *minimal social laws.* These are social laws that minimize the set of restrictions placed upon the agents, while still avoiding inter-agent conflicts. Minimal social laws allow agents to have maximum flexibility during action selection by only disallowing those activities that would prevent other agents from obtaining their goals; thus, they are minimally restrictive.

We propose here that social laws, including even the minimal type described above, can be overly restrictive because agents must adhere to them in all circumstances – even where the possibility of conflict with other agents is extremely low. By insisting that agents avoid any chance of conflict or deadlock when these circumstances are highly unlikely, even minimal social laws may sometimes be overly restrictive and thereby, inherently inefficient. We will refer to this property of a social law as it being *deterministic*. Consider, for example, a domain consisting of a grid traversed by a group of mobile agents. A deterministic social law for this domain might institute traffic regulations to insure that agents never collide or get stuck and to be useful, it would also allow the agents to

In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI'2000)*, pp. 15-21. Austin, Texas. 2000.

reach whichever nodes they needed. However, being deterministic, this social law would be equally applicable to all distributions describing how agents select nodes to visit and how they travel between them.

In this paper, we examine how knowledge of the probability distributions governing agent behavior in MAS can be applied towards more efficiently coordinating them through a *non-deterministic social law*. For example, in the domain above, knowing (or learning) that the agents tend to uniformly select nodes in the grid to visit can drastically improve our ability to coordinate their movement. We note that from a social engineering perspective, this might appear somewhat counter-intuitive. Much of the work in artificial social systems has been motivated through analogy with how human societies function. We institute laws that govern individual behavior and thereby benefit the community as a whole. However, the analogy between agents and people must not be taken too far. For example, vehicular traffic laws in human society need to be easily remembered, and are thus rarely specific to particular distributions and flows of traffic. They are even less likely to be changed dynamically to reflect learned observations. Instead, epiphenomenal approaches are used: highways are constructed that implicitly redirect vehicles, signal light intervals are dynamically varied, and traffic reports are broadcast via radio – all of these are centralized mechanisms to reduce both congestion and the cognitive burden on human drivers. Traffic regulations themselves are essentially inviolate and for good reason – people would find it difficult to drive safely otherwise. However, agents do not share this limitation. There is nothing inherently worrisome in optimizing social laws to better fit the particular MAS they are intended to govern.

We would like to clarify a point that has been somewhat unclear in the social law literature regarding the efficiency of social laws. Social laws are not algorithms – they do not provide a method for accomplishing a particular task. Rather, they are guidelines that specify a class of valid algorithms (or strategies) for solving problems from a particular domain by partitioning the set of possible algorithms into "law-abiding" and "criminal" sets. Social laws are thus not necessarily instructive. Just as traffic laws in human society do not provide directions but simply legislate certain types of behavior in particular situations, social laws maintain a set of constraints that simplify writing and reasoning about algorithms. Therefore, it is not obviously meaningful to speak about a particular social law's efficiency. Instead, what should be considered are the computational and other costs of the best-known algorithms the social law makes realizable. We may then refer to a social law's efficiency solely in this regard. However, others do not always clearly make this discrimination, particularly with social laws so highly constrained and algorithmically formulated they blur the paradigmatic distinction. In referring to their work, we will sometimes find it convenient to ignore this distinction as well. More generally, we will define the notion of a *non-deterministic social law* as one that does not guarantee it is useful in the technical sense given above, although it is highly likely to be for its expected distribution of agent goals and behaviors. We will call *non-deterministic social algorithms* the algorithms that adhere to these laws and only present expected efficiency results regarding them.

In the next section, we discuss the importance of understanding the expected behaviors of the agents – and not simply their goal spaces – while formulating social laws. After this, we examine a traffic domain originally presented in [4]. We formulate a non-deterministic social law for it that is more efficient than its deterministic counterpart. We then present extensive simulation results that demonstrate the efficacy of our approach.

Using Distribution Information

Social laws in MAS do not always provide sufficient information to write efficient control algorithms for the agents. This is not necessarily a limitation of social laws *per se*. However, it indicates the importance of understanding the expected behavior of the agents as a group somewhere in the system's coordination mechanism, whether it be directly incorporated into the system's social laws as we argue in the next section, or instead, into the actual control algorithms for its agents. Even though we are investigating a coordination paradigm that has no centralized controller, there is no reason to insist that individual agents have no knowledge of their expected group behavior.

To better understand this point, it will be useful to first make explicit the role of social laws from a programmer's perspective. A social law for a multi-agent system is designed to give its agents some measure of autonomy and self-government. While it is essential that each agent follow the law, it is of no concern what the agent actually does as long as all of its activities are legal. In other words, social laws make no recommendations as to how agents should spend their time; they simply insure the agents do not unduly interfere with each other. Formulating a useful social law is computationally demanding, and even determining whether one exists for a MAS is in general NP-complete [4]. Therefore, the development of a social law is taken to be an offline practice. However, once a social law is formulated, it can be repeatedly used without further computational expense. This may be contrasted with negotiation protocols, in which computational effort goes into both formulating a protocol and then subsequently negotiating according to it each time is it employed.

After a social law is designed for a system, it is supplied to the agents' programmers, who are then responsible for implementing control algorithms for the agents that obey it. However, without more information about the expected behavior of the other agents in the system, this may be quite difficult to do efficiently. This is because a social law indicates which set of actions is legal in any encountered situation without providing guidance for selecting among

the legal alternatives. It can be difficult do so without additional information. For example, consider the following domain, taken from [1], in which m agents synchronously travel circularly around an n-node ring, with nodes clockwise labeled from 1…n. At each time step, an agent can move to either of its two neighboring nodes or it can remain immobile. A minimal social law presented in [1] that permitted these agents to travel was:

(1) Staying immobile is forbidden if the node that can be reached by a single counterclockwise movement is occupied.
(2) Moving counterclockwise is allowed only if the two nodes that would be encountered by moving counter-clockwise twice are free.

While this social law provides a framework that guarantees two agents cannot collide or deadlock, it does not provide any practical guidance for how to actually move agents around the ring. If an agent is on node k and wishes to travel to node j, $j>k$, should it take a clockwise or counterclockwise path? Supposing $j-k<n/2$, the agent should clearly move clockwise. However, if $j-k>n/2$, it is not obvious which direction is best without knowing both the current value of m and how other agents tend to move (or stay immobile) on the ring. If the agent tried to reach node j but was blocked k steps along the path between them, it might then have to travel clockwise around the ring to j, thereby incurring a $2k$ penalty for its unsuccessful counterclockwise attempt. The primary question here is how far an agent can expect to move counterclockwise without being blocked by another agent. Although we do not further analyze this problem here, this simple example makes clear the need for an individual agent to have available more information than that provided directly by its social law or sensory capabilities.

The Multi-Agent Grid System

We now examine the multi-agent traffic domain presented in [4], which we will refer to as the Grid System. This domain consists of an nxn grid that is traversed by m mobile agents (e.g., robots), as shown in Figure 1. The rows and columns of the grid form lanes that the agents can navigate. We assume that time is discrete and the system is synchronous, so at every time step, each agent is located at some grid coordinate. In this system, agents are given goals in the form of grid coordinates to which they must navigate. Every time an agent reaches its destination, it receives a new goal to visit. For example, the agents might be transporting goods in a warehouse and are alternatively picking up and dropping off items. (We note such systems are currently in frequent commercial use.)

The main consideration here is how to insure that the agents do not collide while they navigate the grid. For example, the most naïve strategy would simply "snake" the agents in a Hamiltonian cycle around the grid, as shown in Figure 2A. Each trip between any two nodes would take $O(n^2)$ time units to complete, which does not compare favorably with the $O(n)$ steps an agent would take to make the same trip in isolation – i.e., with no other agents present. Because this domain does not have cooperative goals – ones that agents work together to achieve – the time an agent would take to complete a task in isolation is the optimum that any social law could achieve.

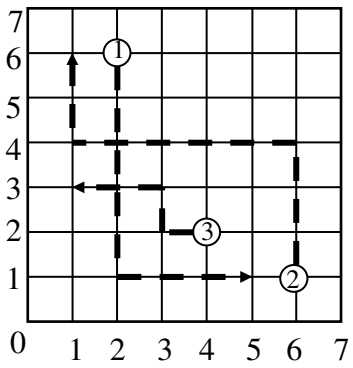


Figure 1 – An 8x8 grid with three agents that will travel along the indicated paths. Notice that in three time steps, agents 1& 3 will collide at coordinate (2,3)

In [4], a complex, deterministic social law is presented for the Grid System that guarantees agents can achieve their goals within certain time bounds. This law requires that agents only use certain rows and columns in the grid for "long distance" travel, much like we use a highway. When an agent reaches the neighborhood of its goal, it then travels to it directly along the "local" grid, as illustrated in Figure 2B.

Summarizing their results, an agent that can achieve a goal in time t in isolation can achieve it using their social law in time $t+2n+o(n)$, assuming $m=O(\sqrt{n})$ and $m<<n$. For the case where $m \le n$, a variant of this law provides that each goal can be achieved in $4n$ time steps.

It is helpful to keep in mind that with $m=O(\sqrt{n})$, the grid is very sparsely populated. For example, if $n=100$, a grid containing 10,000 locations would have on the order of 10 agents moving on it. We are interested in answering the following questions: by insuring generality, is the deterministic social law framework overly constraining? How can its assumptions be loosened in order to achieve a more efficient coordination system? Can we both increase the number of agents travelling on the grid and simultaneously decrease the amount of time they take to reach their goals?

The Uniform Grid System

In this section, we consider the grid system presented above under a particular probability distribution describing an agent's goal selection. Namely, we will assume that the goals are uniformly distributed over all points (x, y) on the grid:

$$\Pr[(x,y)\,is\,a\,goal] = \frac{1}{n^2}, \quad 0 \le x, y < n$$

It is important to note that this assumption will certainly not always be valid, and the non-deterministic social law we present here is not intended for systems where it is not. However, for MAS with agents described by this distribution, we can obtain far more efficient results than those in [4].

Towards determining a lower bound for the non-deterministic social law's efficiency, we first determine the expected distance between two randomly selected integral coordinates, which we call Δ, on a line from [0, n-1] inclusive:

$$E(\Delta) = \sum_{i=1}^{n-1} i \Pr(\Delta = i) = \sum_{i=1}^{n-1} i \frac{2(n-i)}{n^2} = \frac{n}{3} - \frac{1}{3n}$$

On a two-dimensional grid, the expected distance between a pair of successive goals will be $2E(\Delta)$, because the total distance will be the sum of the distances along each axis independently. We will call this value the *isolation time*, denoted by ΔG; it is the expected travel time between goals for an unconstrained, isolated agent. It is therefore also a lower bound on the time taken by any social law governing the uniform grid system. Our goal is to formulate a non-deterministic social law that approaches this lower bound as closely as possible.

Our approach will be to essentially allow the agents to move as they would in isolation. They will explicitly check to make sure their moves are "safe," and take corrective action if necessary. We assume that each agent has sufficient sensory capabilities to realize that other agents are in its immediate vicinity, i.e. up to 2 steps away. In the event a transition between nodes would cause a collision, an agent simply waits to try again on the next move. If an agent is blocked for an extended period along its path, the social law requires that it formulate some alternate route to its destination. Particularly important in this case is ensuring that the deadlock recovery mechanism maintains the assumed probability distribution describing the agents' movements through the grid.

Notice that this approach does *not* guarantee deadlock will be avoided. It is possible (however unlikely) that two agents headed in opposite directions along a column or row can indefinitely block one another, even after repeatedly trying alternate paths to their destinations. In practice, we might try to detect such situations and formulate rules of encounter to avoid them. However, in tens of millions of simulation runs, non-recoverable deadlock has never been encountered. Nonetheless, the non-deterministic social law shown below, which we call law Traffic Law U (for uniform), is not guaranteed in its present form to be useful in the technical sense defined in the introduction:

Traffic Law U

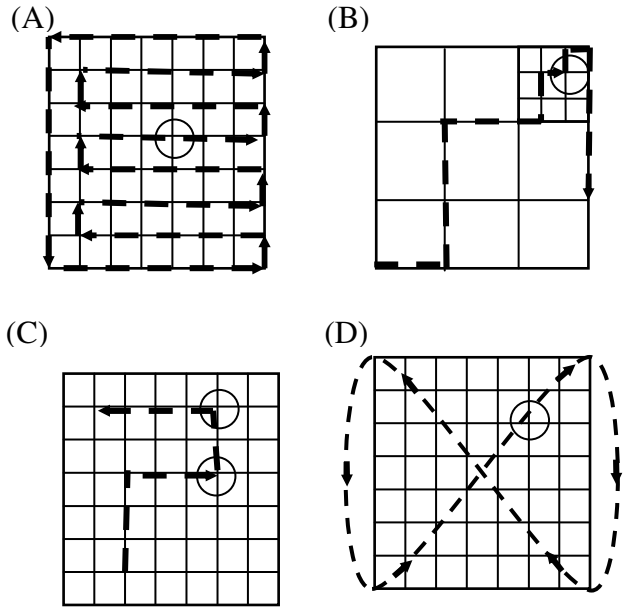


Figure 2 – Different strategies an agent might use to traverse the grid: (A) Walk a Hamiltonian cycle; (B) Navigate through *course* grid until reaching appropriate neighborhood and then use *fine* grid; (C) Take a minimum length path between points; (D) Loop in a "figure 8" path through grid. Circles in the above figures represent goals.

1) At step i+1, an agent may not move to a spot occupied by another agent at step i.
2) If more than one agent simultaneously wants to move to a coordinate, only one, chosen at random, is permitted to do so. The rest must remain where they are and wait one turn before trying again.
3) If an agent has remained immobile for more than *k* turns because its path has been blocked, it must pick another route to its goal.

We will refer to the condition of rule 2 of this law as a *collision* and the requirement of rule 3 as *rerouting*. Rule 1 in the above law is a conservative measure that prevents an agent from moving to a spot most recently occupied by another agent. While technically unnecessary, it allows us to avoid the nightmare of inter-agent communication and coordination that would be necessary for moving an immediately adjacent queue of agents simultaneously. Notice that the Traffic Law U leaves the precise strategy for picking an alternate route in rule 3 unspecified. Any particular implementation of a non-deterministic social algorithm that adheres to Traffic Law U will have to pick some mechanism for selecting this alternative route. This could, for example, involve dynamic negotiation between the agents, random selection, or some other strategy. Below, we examine a method that randomly picks an

intermediate goal to visit along the way to the agent's actual destination in case it gets stuck somewhere.

We define an *L-path* as a path between two grid points that contains at most one turn (i.e., change of direction), so called because of its resemblance to the letter L. (See Figure 2(c).) Points not on the same row or column will have two L-paths connecting them. Otherwise, there will be only one. A *route* is defined to be a sequence of L-paths. We now give a non-deterministic social algorithm that adheres to Traffic Law U:

Algorithm 1:
1. Select a new goal *g*.
2. Let P be a random L-Path from current position to *g*.
3. Set route R = P
4. Move along route R towards *g*, following rules 1 and 2 of Traffic Law U.
5. If blocked for more than *k* steps, do the following:
   a. Randomly select new intermediary goal *g'*
   b. Let P1 be a random L-path from current position to *g'*
   c. Let P2 be a random L-path from *g'* to *g*.
   d. Set R = P1, P2
   e. Go to step 4.
6. Upon reaching goal *g*, go to step 1.

The insistence that agents travel along L-paths is well motivated for maintaining the assumed distribution of agents in the grid. For example, were the agents to travel along random paths (i.e., completely shuffled L-paths), this would induce a normal distribution of the agents, more heavily favoring the center region of the grid and leading to higher numbers of collisions and rerouting. L-paths are to be preferred because they more uniformly distribute the agents and thus, make collisions far less likely. Furthermore, assuming that turning mobile robots requires greater energy than moving them in a straight direction and additionally interferes with dead-reckoning location strategies by introducing additional uncertainty, L-paths are to be preferred for practical, non-distribution specific reasons as well.

Analysis

The efficiency of this algorithm is strictly determined by the number of collisions and amount of rerouting an agent has to do. In the absence of these, each agent would achieve optimal time, because the L-path to its goal is a shortest length route to it. However, in the presence of other agents, both collisions and rerouting are inevitable and can incur prohibitive time penalties. With respect to each agent, a collision has cost 1 because of the incurred delay. Rerouting has cost of at least $2\Delta G$, because deadlock may occur during the rerouting process itself. However, there is no recursive rerouting – the agent simply reroutes with respect to the original goal, not the intermediary selected in step (5a) of the algorithm.

We will first provide a loose upper bound to the expected running time for an agent to travel between successive goals on an *n*x*n* grid containing *m* agents. We use this to determine how many agents can be allowed on the grid simultaneously given how much overhead (i.e. wasted travel time) is acceptable. We then present extensive simulation results for Algorithm 1, due to the difficultly of obtaining tighter bounds for its running time.

Analytic Results

To determine how many agents can simultaneously traverse the grid without incurring unreasonable delays due to congestion, we approximately model an agent's movement through the Uniform Grid System as if it were governed by a negative binomial distribution. This approximation will become increasingly inaccurate in systems where the grid is more heavily congested, in which case we must turn to the simulation results given below.

For Algorithm 1, we bound E_t, the expected travel time between goals as:

$$E_t \le E(\text{ time moving towards goal }) + E(\text{ time recovering from deadlock })$$

We define the probabilities of colliding and successful transitions as P_c and P_s respectively:

$$P_c = \frac{m-1}{n^2-1}, \quad P_s = 1 - P_c$$

Note that P_c would be seem to be double the given value but we assume that half the time an agent is involved in a potential collision, it is the one selected to move per rule 2 of Traffic Law U, and no time penalty is thereby incurred. We bound the probability of deadlock P_d by considering that it occurs only when agents collide and then subsequently block each other. Separately accounting for interior and border regions, we have:

$$P_d = \frac{(n^2-4n)}{n^2}\frac{P_c}{4} + \frac{4n}{n^2}\frac{P_c}{3} = \frac{P_c}{4} + \frac{P_c}{3n} \approx \frac{P_c}{4}$$

Recall ΔG, the isolation time, is given by:

$$\Delta G = 2E(\Delta) = \frac{2n}{3} - \frac{2}{3n} \approx 2n/3$$

We calculate E_{goal}, the expected time an agent spends moving towards its goal using our negative binomial distribution assumption:

$$E_{goal} = \frac{\Delta G}{P_s}$$

We determine $E_{deadlock}$, the expected time an agent spends recovering from deadlocks, explicitly noting that the agent may deadlock in the midst of deadlock recovery:

$$E_{deadlock} \le \left(\frac{\Delta G}{P_s}\right) P_d (2\Delta G + P_d (2\Delta G + P_d (2\Delta G + ...)...)$$
$$= \left(\frac{\Delta G}{P_s}\right) 2\Delta G \frac{P_d}{1 - P_d}$$

We then have the expected time between successive goals is:

$$E_t \le \frac{\Delta G}{P_s} + \left(\frac{\Delta G}{P_s}\right) 2\Delta G \frac{P_d}{1 - P_d}$$

Next, we define c^*, the ratio between the expected and isolation times when traveling between successive goals. It is a measure of the overhead due to agent interaction while traversing the grid:

$$c^* = \frac{E_t}{\Delta G} \le \frac{1}{P_s} + \left(\frac{2\Delta G}{P_s}\right) \frac{P_d}{1 - P_d} \le \frac{1}{1 - P_c} + \left(\frac{2\Delta G}{1 - P_c}\right) \frac{P_c / 4}{1 - P_c / 4}$$

Recalling the above definition of P_c, we solve for the number of agents m as a function of c^* and n:

$$m \approx \frac{c^* - 1}{c^* + \Delta G/2}(n^2) + 1 = \frac{c^* - 1}{c^* + n/3}(n^2) + 1$$

We now have a handle on how many agents can be allowed onto an nxn grid given some level of acceptable overhead c^*. For example, on a 100x100 grid, if it is acceptable for an agent to spend 1.3 times longer between successive goals than it would on the grid alone, then we expect that roughly 87 agents can be permitted onto the grid simultaneously. Note that this is actually an underestimate because of the non-tight bound for E_t determined above. The actual number demonstrated in simulation for $c^* = 1.3$ is m=n, or in this case, m=100.

Simulation Results

A Java-based simulator was written for the Uniform Grid System employing Traffic Law U and Algorithm 1. Our approach for each grid of size n was to slowly increase the number of agents, m, observing how this impacted the average time of an agent to achieve its goals. We first consider the case where $m = c\sqrt{n}$. As expected, the time taken for an agent to achieve its goals on average is essentially equal to its isolation time. Tables 1 and 2 contain the cases for c = 1 and 10 respectively.

Each simulation was run until the agents globally achieved 10,000 goals. In the table: *#S* represents the number of time steps simulated; *CP* is the total collision penalty for the simulation; *RP* is the total rerouting penalty; *Avg* is the average time an agent took to achieve a goal; *ΔG* is the time an agent would ideally take in isolation; *c** is *Avg/ΔG;* and *%+* is 100x*(Avg –ΔG)/ΔG*. We note that lower *c** values are better, and a value of 1 is the best that can be achieved by any social law in this domain.

We then examined cases where $m = cn$, where $1 \le c < n$. As c approaches n, the density of the agents increases to the point where they become hopelessly crowded, and navigation becomes extraordinarily inefficient. As this happens, it becomes more efficient to simply "snake" the agents around the grid in a Hamiltonian cycle as described above. Graphs 1 and 2 display the rate of change in *c** (=*Avg/ΔG*) as a function of c (=m/n) for n=10 and 100 respectively.

Finally, we examine our results for the case where c = 1 (m=n), where we find that empirically, c* is roughly around 4/3 for all values of n.

Table 1: $m = \sqrt{n}$

n	m	#S	CP	RP	Avg	ΔG	c*	%+
10	3	23728	3478	5069	7.12	6.6	1.08	7.85
20	4	34445	2562	7574	13.78	13.3	1.04	3.59
100	10	6589	192	2755	65.89	66.66	0.99	-1.16
200	14	9737	70	1624	136.32	133.33	1.02	2.24
500	22	15208	59	3229	334.58	333.33	1.00	0.37
1000	32	21365	22	2227	683.68	666.66	1.03	2.55

Table 2: $m = 10\sqrt{n}$

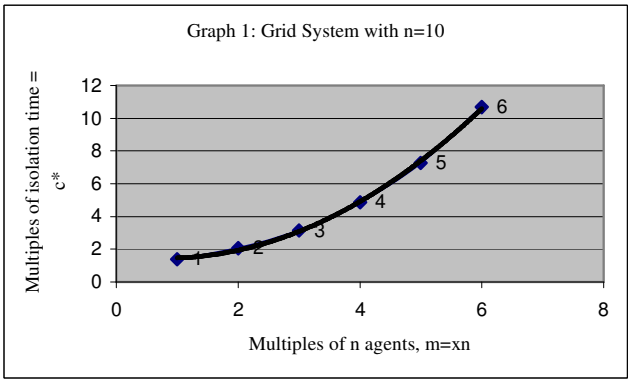
n	m	#S	CP	RP	Avg	ΔG	c*	%+
10	30	724	13354	5582	21.7	6.60	3.29	2.29
20	40	542	5146	8282	21.61	13.30	1.62	0.62
50	70	590	2708	13775	41.3	33.32	1.24	0.24
100	100	763	1634	17239	76.3	66.66	1.14	0.14
500	220	1752	643	40440	385.1	333.33	1.16	0.16
1000	320	2617	460	55176	836.6	666.67	1.25	0.25

Table 3: $m = n$

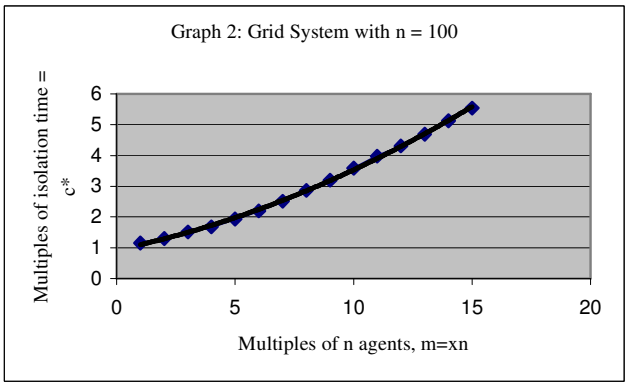
n	#S	CP	RP	Avg	ΔG	c*	%+
10	878	1758	1977	8.78	6.60	1.33	0.33
20	846	1895	4361	16.92	13.30	1.27	0.27
50	815	2363	12609	40.71	33.32	1.22	0.22
100	791	1960	22494	78.94	66.66	1.18	0.18
500	825	1376	81594	412.09	333.33	1.24	0.24
1000	931	1139	152955	929.14	666.67	1.39	0.39

Comparison of results

Algorithm 1 is near optimal in the logarithmic cases shown in Tables 1 and 2, where $m = c\sqrt{n}$. Only in the case



where $c \geq m/2$ does the performance degrade



substantially. When $m = cn$, we observe a near constant multiplicative cost of approximately 1.3ΔG for $c = 1$. As c starts to increase, we note the expected penalty observed in the average time it takes an agent to reach its goal. Finally, as c approaches n itself, the number of agents approaches n^2, and it would be best to dynamically switch to the Hamiltonian path strategy. In the table below, we compare the expected time for an agent to reach its goal in our approach and the one taken in [4]:

Table 4: Comparison of non-deterministic Algorithm 1 with the deterministic social law presented in [4]:

m =	Expected Time to Goal	
	Non-Deterministic	Deterministic
$\sqrt{n}$	ΔG (=2n/3)	cΔG, c>2
$10\sqrt{n}$	Approaches ΔG	cΔG, c>2
n	1.3ΔG=13n/15	4n
cn	See graphs	Not applicable

Conclusions

In this paper, we proposed that general purpose, deterministic social laws appropriate for all circumstances may be inappropriate for the situations MAS actually encounter. In particular, we argued that knowledge of the underlying distributions describing agent behavior can give us new ways of coordinating MAS and help us formulate more efficient social laws. We demonstrated this by revisiting a previously studied traffic domain problem. By assuming a particular distribution of both agent goals and their deadlock recovery behavior, we were able to formulate a simple and more efficient strategy for coordinating the movement of agents throughout the grid.

Future work in this domain includes more precisely characterizing the runtime complexity of Algorithm 1, exploring how well the system works when faced with other distributions, i.e., how sensitive this formulation is to the actual encountered behavior, and exploring other coordination domains that might be amenable to this approach.

Acknowledgements

This material is based upon work supported by the Advanced Research Projects Agency of the Department of Defense under contract number F30602—94—C—0204, monitored through Rome Laboratory. Special thanks to D. Fitoussi and L. Weisman.

References

[1] Fitoussi, D. and Tennenholtz, M. Minimal Social Laws. In Proc. Of the Fifteenth National Conference on Artificial Intelligence, p26-31. 1998.

[2] Moses, Y., and Tennenholtz, M. Artificial Social Systems. *Computers and Artificial Intelligence*. 14(6):533-562.

[3] Rosenschein, J.S., and Zlotkin, G. Rules of Encounter: Design Conventions for Automated Negotiation among Computers. MIT Press. 1994.

[4] Shoham, Y., and Tennenholtz, M. Social Laws for Artificial Agent Societies: Off-line Design. *Artificial Intelligence* 73. 1995.

[5] Stuart, C. An Implementation of a Multi-Agent Plan Synchronizer. In Proc. Ninth International Joint Conference on Artificial Intelligence. 1985.