Generating New Regulations by Learning from Experience

Jan Koeppen
University of Barcelona
Gran Via 585 08007 Barcelona, Spain
janfrancisco@gmx.net

Maite Lopez-Sanchez
WAI, MAiA, University of Barcelona
Gran Via 585, 08007 Barcelona, Spain
maite_lopez@ub.edu

ABSTRACT

Both human and multi-agent societies are prone to best function with the inclusion of regulations. Human societies have developed jurisprudence as the theory and philosophy of law. Within it, utilitarianism has the view that laws should be crafted so as to produce the best consequences. Following this same objective, we propose an approach to enhance a multi-agent system with a regulatory authority that generates new regulations –norms– based on the outcome of previous experiences. These regulations are learned by applying a machine learning technique (CBR) that uses previous experiences to solve new problems. As a scenario to evaluate this innovative proposal, we use a simplified version of a traffic simulation scenario, where agents move within a road junction. Gathered experiences can then be easily mapped into regular traffic rules that, if followed, happen to be effective in avoiding undesired situations —and promoting desired ones. Thus, we can conclude that our approach can be successfully used to create new regulations for those multi-agent systems that accomplish two general conditions: to be able to continuously gather and evaluate experiences from its regular functioning; and to be characterized in such a way that similar social situations require similar regulations.

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Distributed Artificial IntelligenceMultiagent systems

General Terms

Algorithms, Experimentation

Keywords

Normative systems, Learning, Agent based simulation, Self-organisation, Norm generation.

1. INTRODUCTION

Regulations have been proven to be useful in both human and multi-agent societies. Human societies use regulations within their legal systems. In fact, they have developed Jurisprudence as the theory and philosophy of law, which tries to obtain a deeper understanding of general issues such as the nature of law, of legal reasoning, or of legal institutions[1]. Within it, Normative Jurisprudence is concerned with normative or evaluative theories of law. It tries to answer questions such as "What is the purpose of law? or What sorts of acts should be subject to punishment?. Normative Jurisprudence has different schools. Among them, Deontology [7] can be described as an ethical theory concerned with duties and rights. On the other hand, Utilitarianism [12] takes the view that the laws should be crafted so as to produce the best consequences. When translating these approaches from human societies to MAS societies, it is obvious that a large number of simplifications have to be taken. Nevertheless, we think that it is still possible to keep and combine their fundamental objectives: to define specific prohibitions, permissions and obligations that promote desired overall system's behaviour for a given MAS society. Thus, the aim of this paper is to define a computational mechanism able to synthesize norms that succeed in the proper regulation of multi-agent societies[2].

We approach this regulation generation problem by learning from the experience of on-going activities within the MAS society. We have chosen Case-Based Reasoning (CBR) as the learning technique to apply. Briefly, CBR solves new problems –i.e., cases– by adapting the solution of similar problems from the knowledge base (which is a compound of solved problems). The selection of this learning technique is somehow inspired in the Anglo-American common law tradition, where judges use legal precedents to make decisions. Hence, using our terminology, we can interpret that judges re*solve* legal *cases* based on the way similar *cases* were previously re*solved*. More specifically, our approach defines a case as a compound of a problem –i.e., a social situation or context– and its associated solution, which in our case corresponds to the regulations that are applied in those contexts. In this manner, the overall learning objective becomes to define cases whose application leads to desired social situations. In CBR, problem description is key, and therefore, we have tested different problem representations that consider global and partial scopes. On the other hand, CBR is a supervised learning method that requires an expert to provide the system with correct problem solutions. Nevertheless, we want to generate best regulations without external knowledge, and thus, CBR cannot be directly applied. Instead, we propose to include an exploratory pseudo-random approach so that CBR becomes unsupervised.

Rather than by individual agents in the society, we assume learning to be performed by an independent regulatory authority within the MAS, able to observe and establish its

[1]Jurisprudence definition extracted from Black's Law Dictionary: http://www.blackslawdictionary.com

[2]We assume goals act as a reference that does not evolve.

norms. Therefore, we are taking an organizational centered perspective over the MAS as opposed to an agent-centered perspective. The underlying rationale is to restrict the focus of our research. An organizational point of view allows to have learning devoted to finding the best regulations for a whole society and to do it while interactions are taking place. On the contrary, taking an individual centered approach –where learning is performed by individual selfish agents– would also require considering additional aspects such as agreement, trust, uncertainty or communication.

The paper is structured as follows: next section introduces related work. Section 3 describes the tested scenario, section 4 details the learning process, and subsequent section 5 presents its empirical evaluation. Finally, some conclusions and future work are drawn in last Section 6.

2. RELATED WORK

Although Artificial Intelligence and Law have been related since a first article from McCarty [11], related research is not usually concerned with machine learning. This is less the case within the MAS area, where some learning techniques have been successfully applied. In fact, Multi-Agent Reinforcement Leaning [4] is quite widely used for individual agent learning. Nevertheless its usage is much more scarce for organizational centered approaches, where an exception is the work by Zhang et al.[19] devoted to improve system's organization. Our work uses CBR as an alternative learning technique, which is also based on system experience, but results in clearer knowledge representations —i.e., cases.

On the other hand, research on norms in multi-agent systems is a quite active area. Just to mention a few works: Boella and van der Torre have done relevant contributions [3] in norm characterization; Campos et al. [5] have proposed norm adaptation methods to specific network scenarios; Artikis et al.[2] have studied the definition of dynamic social conventions (protocols); and Savarimuthu et al. [16] as well as Kota et al. [10] work on norm emergence. Within this area, norm generation has been studied less frequently. Shoham and Tennenholtz [17] focus on norm synthesis by considering a state transition system: they explore the state-space enumeration and state it is NP-complete through a reduction from 3-SAT. Similarly, Hoek et al. [18] synthesize social laws as a model checking problem –again NP-Complete– that requires a complete action-based alternative transition system representation. In our case, CBR has the advantage that, although cases represent the search space, they do not need to be exhaustive, since they can be representatives of a set of similar problems requiring similar solutions. Furthermore, our approach is applied at run-time, being able to generate new norms during the execution of the system (this has the additional advantage of adapting to new situations). An intermediate approach is this of Christelis and Rovatsos [6], that synthesize generalized norms over general state specifications in planning domains. These domains allow for a local search around declarative specifications of states using planning AI methods. From our point of view, CBR allows the application to a wider range of domains, in particular to those where (i) experiences can be continuously gathered and evaluated, and where (ii) similar social situations require similar regulations (i.e., the continuity solution assumption).

Regarding implementation issues, it might be worth mentioning a related work on system monitoring by Modgil et al.[13] which is able to recognize norm compliance; and another one on traffic domain by Dunkel et al. [8] devoted to managing traffic systems. We have also used a simplified traffic scenario to test our innovative approach empirically.

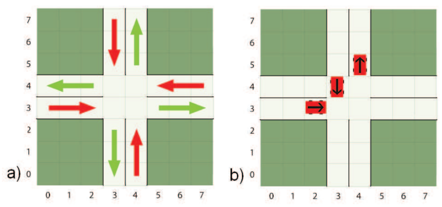


Figure 1: Orthogonal road junction: a) feeder and exit lines, b)traveling cars.

3. TRAFFIC SCENARIO

In order to test our learning approach, we have chosen a simplification of a traffic scenario. It has been developed as a multi-agent based simulation model in Repast [14]. This traffic scenario is an orthogonal two-road junction, where car agents travel along roads towards different destinations. As figure 1 shows, the environment has been discretized by means of a square grid whose cells have the size of a car. Gray (central) cells represent roads and green (corner) cells correspond to their surrounding non-transitable fields. Each road lane has a direction of traffic. Agents can join the road from four different entrance points –i.e., four incoming or feeder lanes (see left side of Figure 1)– and choose the exit point, so they decide the route to follow. Time, measured in ticks, is also discrete. Moreover, cars do have constant speed, so they can only move to adjacent cells in a single tick. Agent possible actions are stop, move forward, turn right, or turn left. Nevertheless, cars just turn in the intersection area and always obey the rules of right side traffic (i.e. they turn right in the first cell of the intersection whereas left turnings require to further traverse the junction and turn on the second cell). Furthermore, car agents also follow the social norms described in section 4 by stopping or moving whenever required.

4. NORM GENERATION THROUGH CASE-BASED REASONING

Multi-agent systems have been enriched with different regulations –norms, constraints, protocols, etc– with the aim of better organizing the society by restricting both individual behaviours and the way interactions are performed. In general, regulated societies build their norms as an implicit common agreement, assuming most of their individuals will respect them. Regulations can come from a norm emergence process or by having a regulatory authority dictating them. Furthermore, they can be created based on previous experiences or by anticipating situations that may appear. Nevertheless, since the number of possible outcomes of complex systems is so large, most societies regulate just those situations that have already occurred so far. This paper focuses on those regulations that can be established based on the experience of the regular functioning of MAS societies. We assume these societies have regulatory authorities that gather experiences in an on-going basis. Inspired in jurisprudence used in the Anglo-American common law tradition, we have enriched our MAS with a case based regulatory system. It is is charge of analyzing previous experiences and deciding what (if any) regulations should be applied for specific situation contexts in order to avoid undesired outcomes.

In order to do it, a regulatory authority must be able to first define the goals whose accomplishment guarantees system's performance or its overall desired behaviour. In our traffic scenario, the main goal is to minimize the number of collisions whilst keeping a fluid traffic. This is so because, obviously, if all cars stop, then there will be no collisions at all but cars will not accomplish their individual goals —which most probably will include reaching their destinations. Therefore, we are making an underlying assumption that is that social regulations should guarantee individuals to have enough autonomy so to accomplish their individual goals. Otherwise, punishments should be included to promote norm compliance. In summary, we can somehow interpret that the regulatory authority tries to guarantee basic common agreement about the norms it establishes.

Second, the regulatory authority must have the ability to observe the society in a way that it is able to identify undesired situations —that is, situations where goals are not being accomplished. In our traffic case, both collisions and blockages are main undesired situations.

Afterwards, the regulatory authority should be able to propose regulations that try to prevent undesired situations from being repeated in the future. Prohibitions should be done over those agents' actions that lead to undesired situations. Analogously, obligations can be used to promote desired actions. For example, if we consider our traffic junction, if there is a collision because two cars run on each other, then it is possible to propose a new regulation that prohibits cars to move when they happen to be in the same situation. On the other hand, if no collisions happen when cars traverse the junction it may be useful to create the obligation of keeping moving to prevent blockages. Obviously, deciding which actions should be prohibited or obliged is not a straightforward decision, and that is the reason we introduce automatic learning into the process.

Finally, whenever a new regulation is created and applied on the multi-agent system, the learning process requires the analysis of the consequences of its application. Thus, we need the regulatory authority to observe the society's evolution and to label the experience of applying this new regulation with its subsequent outcome. In this manner, regulatory knowledge is refined in an on-going basis.

The remaining of this section provides further details of our proposed approach. First subsection specifies the architecture of the MAS applied to the traffic scenario, and subsequent subsections detail the learning process.

4.1 Architecture

Following an organizational centered approach, we assume that the multi-agent system in our traffic scenario consists of a set of external agents that interact within a road environment together with a regulatory authority (see Figure 2). External agents play a car role; they are able to observe other car agents and to perform certain actions such as join, traverse, and leave the environment. Regarding the regulatory authority, its aim is to promote fluid car traffic flow with as few as possible collisions amongst traffic participants. This authority is constituted by staff permanent

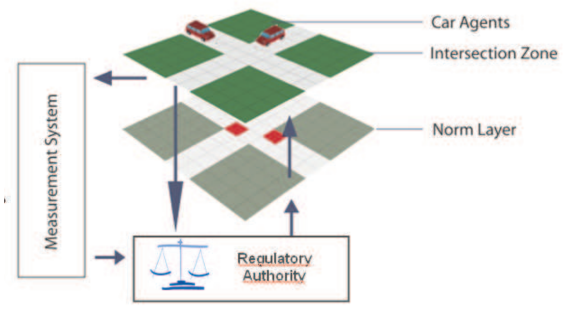


Figure 2: Traffic scenario architecture.

agents that perform regulation tasks. From those agents, we highlight the one in charge of defining current norms –we call it norm agent– and the one conducting the learning process —the CBR agent. Nevertheless, there are other staff agents that provide infrastructure services, such as the ones in the tracking system, in charge of obtaining information from the environment; the scene manager, in charge of runtime details; or the monitoring agents, which provide statistical analysis of the overall system operation.

The norm agent uses the regulatory knowledge from the CBR agent to specify the traffic rules that will be applied in the road environment. As a result, it updates a norm layer that is publicly available for the car agents so they become aware of the norms and can thus follow them. Agents conduct this norm updating process continuously, creating new norms when required or applying previously existing ones. The CBR agent will be the one in charge of taking this last decision. Next subsection details how it is performed.

4.2 Unsupervised CBR cycle

Case-based reasoning is a technique that solves new problems based on past experiences [1]. Experiences are stored in the form of cases, where a case is a description of a problem and its possible solution $Case = \langle probl, sol \rangle$. Cases are stored and maintained in a knowledge base (or case base) for further usage. Briefly, when a new problem is encountered (and thus, it lacks a solution), the CBR process searches for the most similar problem in the case base and adapts its associated solution to solve the current problem. The description of the target problem, together with the provided solution and related information about its performance, constitute a new case that can be in turn stored in the case base. Case performance –i.e., how well the derived solution solved the problem– depends on the continuity of the domain or, in other words, if for the domain it holds that similar problems require similar solutions. This overall process is usually explained in terms of what it is known to be the CBR cycle. It is characterized by four different steps: retrieve, reuse, revise and retain. Before describing them, it is worth mentioning that a case for us is composed of a traffic situation –car distribution–, the regulations –move /not move– that should be applied in such traffic context, and a case performance measure (see subsection 4.3 for further details).

Retrieve: Given a traffic situation description, we first retrieve from our knowledge base the case that is most relevant to solve it. Relevance here is interpreted as similarity, and thus, we search for a case that describes the most similar traffic situation. More specifically, as we will see in next subsection 4.3 a case is considered to be similar to another if it represents the same number of cars and if these cars are located at rotationally equivalent cells. The retrieved case will include the regulations that were applied for its traffic situation and a score of its application.

Standard CBR systems are considered as supervised learning methods because they assume there is a pre-existing knowledge base, or that at least, a supervisor can provide solutions for new cases to be learned. Nevertheless, we face an unsupervised learning scenario, since we lack the necessary knowledge to determine the proper traffic rules that should be applied for specific situations. Therefore, it can well be the case that the retrieve phase does not provide any case. In fact, we encounter this situation right at the beginning, since we still lack experience. Hence, if no case has been retrieved, we need to somehow generate a new solution by exploring the space of possible solutions, which in our case means to try different combinations of traffic restrictions (norms). In our current implementation, exploration is performed by randomly assigning stopping/moving restrictions to those cells having cars (avoiding empty cells is an heuristic that prunes the search space). Furthermore, since this pseudo-random solution may not be optimal, we extend the cases to include several alternate solutions (generated in the same way) with a performance measure associated to each of them. The number of possible solutions is bounded in order to differentiate a learning phase –when alternate solutions are built– from a subsequent testing phase —when the case is considered to be learned (i.e., closed) and is applied without adding new solutions. Obviously, this limit in the number of explored solutions prevents us from guaranteeing optimal solutions, but they can still be useful to accomplish the goals of our regulatory authority. Powell et al.[15] have a similar approach to unsupervised CBR that uses reinforcement learning.

In the **reuse** phase, the solution of the retrieved case is mapped to the target problem. This may involve adapting the solution as needed to fit the new situation. In our case, since a case may have more than one associated solution, the one having the best performance results is the one chosen. Reuse is done afterwards by translating the traffic rules of the chosen solution to locations in the new solution that may be rotated if the target problem is a rotated version of the retrieved case.

Afterwards, having mapped the previous solution to the target situation, test the new solution and, if necessary, **revise**. In our traffic scenario this means to dictate the traffic norms to car agents (see previous subsection 4.1), and to observe the outcome of their application in the simulation. In current implementation, the regulatory authority checks if goals are fulfilled by observing next[3] simulation step (tick). Then, it updates the performance measure based on the number of resulting collisions and the number of applied prohibition rules: in order to promote fluid traffic, it penalizes over-regulated solutions —i.e., those abusing from preventing the cars from moving. Although system's goals are two-folded –collision avoidance and fluid traffic– they may have different relevance and, therefore, we use a weighted performance updating formula.

[3]Different time intervals could be used depending on the delay of norm application effects.

Finally, the cycle ends with the **retain** phase, that consists on the storage the resulting experience in the knowledge base. In our unsupervised CBR scenario this may lead to three different possibilities: i) If a new case was generated, then it will be stored in the case base; ii) If an existing case was retrieved and a new solution for it was generated, then retain becomes an update of the current case; and iii) if the retrieved case was closed –and thus, no solutions were added– the only required update is the performance measure[4]. This will allow the CBR agent to choose among different traffic rules depending on their application outcome. It is worth noticing that for non-deterministic environments, a desirable regulation may become undesirable further in time and become desirable again under changing circumstances. As we can see, this last step enriches the set of stored experiences, and thus it better prepares the system for future encountered problems as far as they satisfy the underlying premise that similar problems have similar solutions.

4.3 Cases and Norms

As we have already mentioned, a case in CBR is generally understood as the description of a problem and its associated solution: $Case = \langle probl, sol \rangle$ where $prob \in StateSpace$ and $sol \in Norms$. Taking into account our traffic domain, a problem description represents one particular traffic situation whereas the solution corresponds to the traffic rules that should be applied for this particular context. The regulatory authority describes traffic situations in terms of the information it gathers from the system (see section 3) : empty and occupied cells, and the headings of those cars located at occupied cells. Traffic situations can be described by considering a global point of view or a local perspective. A global scope in the representation will imply a large area of the environment and will contain all cars in the environment, no matter their location. On the other hand, a local perspective is focused in a narrower environment area and thus, only those cars near the reference point will be considered. A global scope has the advantage that it represents a complete knowledge but the disadvantage of implying a large search space. Regarding the partial scope, although being smaller in its representation size –and thus, search space–, it may fail in representing some important pieces of knowledge. Therefore, as both approaches present pros and cons, we have modeled them both in our particular traffic scenario (see next evaluation section 5 for a comparison). The remaining of this subsection presents them and the associated solutions (norms) they have within our case representation. In fact, depending on the considered scope, norms will be applied to all involved agents –if global scope– or just to the single agent that is acting as reference in the partial representation.

4.3.1 Global scope

When representing a complete traffic situation, the number of possible distributions of cars in the environment becomes high even if just considering the 7×7 example grid in Figure 1. Nevertheless, some simplifications can be taken. First, by assuming that car agents do have basic driving skills it is possible to reduce the size of the environment grid down to the intersection zone (see left side in Figure 3). These skills correspond to basic capacities such as planning

[4]Additionally, for all three possibilities we also store/update how many times the case has been applied.

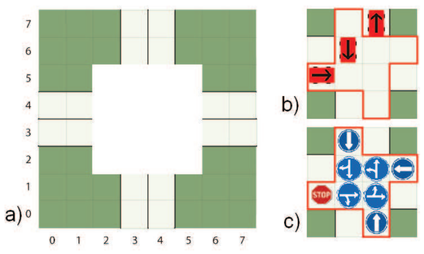


Figure 3: Global scope junction representation: a) initially discarded cells; b) orthogonal shape representing the problem; and c) applied traffic rules.

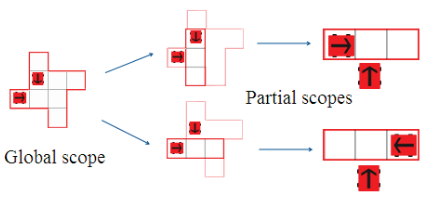


Figure 4: Case global and partial scopes.

a path towards a chosen destination, following this route without leaving the proper road lanes or stopping if a car in front of them in the lane brakes suddenly[5]. Thus, traffic in the feeder and exit lanes (see Figure 1) can be discarded without losing any relevant information. Figure 3 shows how, focusing further on the intersection zone, there are still some cells that can be obviated. These cells correspond to both the field area and the exit lanes, which do not interfere in our simplified traffic. In this manner, the final problem representation can be reduced to 8 cells in the junction area.

The state space ($StateSpace$) we are representing consists thus in 8 cells that can be either empty or occupied by one or several cars. Having more than one car in a cell means a collision. Cars in our simulation are removed when colliding, so there is no need to represent this situation (further details can be found in [9]). Furthermore, a car in an occupied cell can have different headings, but due to the traffic flow restrictions, it will only be one for the cells at the junction entrance (the ones of the feeder lanes) or two for the intersection, since two different traffic directions are allowed there. Overall, we have 4 cells with two possible states –i.e., empty or occupied with a fixed heading– and 4 with 3 possible states –empty and occupied with two alternative headings– so that we have $2^4 * 3^4 = 1296$ different possible traffic situations. Finally, we can have situations that represent the same if we apply the appropriated rotation in their representations. Thus, we can further reduce the state space to $1296/4 = 324$ combinations.

Regarding the associated solution ($sol \in Norms$), it represents the same grid area than the problem (see Figure 3 right down) and for each cell, it has a norm that specifies if the car in this location should stop or should keep moving. From a deontic perspective, these traffic rules are represented, respectively, as the obligation of stopping and the prohibition to stop. Thus, the norm agent first considers the solution provided by the CBR agent (see section 4.1) and, afterwards, it applies traffic signs that can be either the stop sign or a direction sign —whose specific direction will correspond to the one of the road cell.

Finally, as we have mentioned, we lack the optimal solution ($sol \in Norms$) for each problem ($prob \in StateSpace$) and thus, the learning algorithm explores different candidate solutions. Thus, a case in our global scope representation corresponds in fact to $Case = \langle probl, \{(sol, score)\}\rangle$, where for each problem we have a set of solution-score pairs, and where a solution is a combination of traffic rules and it is associated to information about their application outcome.

[5]These basic skills may also be modeled as a set of basic norms, but from our point of view regulations should leave some decisions to the agents, whose autonomy can be regulated but should not be overconstrained.

4.3.2 *Partial scope*

As an alternative to use global information, it is also possible to represent situations centered in the point of view of a single agent. Common agent individual perspectives also imply having a limited observation range. Thus, the partial scope reduces the observation area to a subgrid in front of the reference car. Figure 4 illustrates an example that compares the conceptualization of both scopes: for a given global situation at a certain time step we will have as many partial descriptions as involved agents are. Thus following the example in the figure, two different situations –i.e. $prob1, prob2 \in PartialStateSpace$– will be derived. This, in terms of the CBR learning process, means that they will result in two target cases to solve, and therefore, the CBR process will be invoked twice.

As before, problems ($prob \in PartialStateSpace$) are represented by considering empty and occupied cells. The only differences are that their orientation is relative to the reference car and its shape and dimensions, which do not include the cell containing the reference car, are smaller than the global problem representation. Following previous example, we have a rotated 3×1 sub-grid. There, cell states can be 4 (empty, car forward movement, car left turning, and car right turning) for those two cells corresponding to the inner junction area and 2 possible states (empty cell or occupied with a car moving forward) for the single cell in the junction entrance. Obviously, having a sub-grid implies a smaller state space ($|PartialStateSpace| < |StateSpace|$) and thus, the number of possible cases to handle is much smaller ($4^2 * 2 = 32$ in the example). Our implementation allows the definition of different sight range or subgrids –they are treated as masks over the agent's visibility area– so that they can be empirically studied.

Once we have defined a problem ($prob \in PartialStateSpace$), its solution corresponds to the norms that will be applied to the reference agent. In this manner, cases in our traffic scenario will have a predefined set of two possible traffic rules: the obligation to stop ($obl(stop)$) and the obligation to keep moving following the road traffic direction ($proh(stop)$, so that we have a reduced set of norms: Norms=$\{obl(stop), proh(stop)\}$). Cases in this approach will have a predefined set of two possible solutions and their associated outcome measure $Case = \langle probl, \{(obl(stop), scoreStop), (proh(stop), scoreMove)\}\rangle$ and the main learning task will be to change the score associated to the performance of the application of both rules ($scoreStop$ and $scoreMove$).

4.3.3 Related metrics

Case retrieve and case update phases in our CBR cycle require the specification of two measures: the distance between two cases and the score of associated solutions.

Both global and local approaches compute case distance by comparing every cell in the area (both compared grids have the same size and shape). Differences between two cells $c_i, c_j \in grid$ are considered to be 1 if their occupancy state is different:

$dist(c_i, c_j) = 1$ if $state(c_i) \neq state(c_j)$, where

$state(c_k) = \{empty, occupied\_forward, occupied\_right{-}\_turn, occupied\_left\_turn\}$ and $c_i, c_j, c_k \in grid$

$distance(grid_1, grid_2) = \sum_{c_i \in grid_1, c_j \in grid_2, i=j} dist(c_i, c_j)$

Thus, for example, if $state(c_i) = occupied\_forward$ and $state(c_j) = empty$, then $dist(c_i, c_j) = 1$ and the same distance results if they are occupied with cars with different headings: $state(c_i) = occupied\_forward$ and $state(c_k) = occupied\_right\_turn$ (then $dist(c_i, c_k) = 1$).

The retrieval phase looks for the most similar case in the knowledge base. In our case, the chosen case will be the one for which, if we apply a proper rotation to the retrieved grid, we get a zero distance result when comparing with the grid representing the target problem. Formally: $retrieved\_case = arg\, distance(rotation(grid, \alpha), target\_grid) = 0$ where $\alpha \in \{0, 90, 180, 279\}$ degrees in our orthogonal environment and $grid$ is the representation of the problem component in the case.

Regarding the scoring computation, we have already said that given a retrieved case with different solutions, the norm agent in the regulatory authority will choose the solution with best application performance. In the global scope, this score update is computed by punishing both the number of collisions (n_col) occurred during the next time step in the simulation; and the number of stop traffic rules ($obl(stop)$) that were applied (n_stop). Both measures are accordingly weighted so that we have:

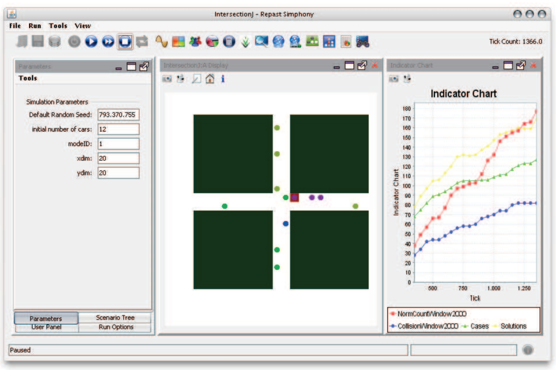
$$global\_score = previous\_global\_score - (w_{col} \cdot n\_col + w_{stop} \cdot n\_stop)$$

Weight values depend on the priority over goals that the regulatory authority has. Our current implementation considers $w_{col} = 5$ and $w_{stop} = 1$ (i.e., a 1 to 5 ratio in the importance of collisions and traffic jams).

Finally, the computation of the partial scope score has to take into account that partial information may lead to different outcomes when applying the same norms to the same partial problem description. In order to deal with this non-deterministic phenomena, we average current with previous outcomes so to smooth the updating effect. Our testing simulation environment allows the definition of several methods, such as, for example, implementing a sliding window over the experience history.

5. EMPIRICAL EVALUATION

As we have previously mentioned, we have performed an empirical evaluation of our proposal about regulation generation by developing a multi-agent based simulation of a traffic road junction scenario. The simulator has been implemented over Repast simphony [14] so that its runtime environment interface can be used to enhance the user interface of our simulator. Figure 5 shows the user interface: top toolbar includes the standard simulation buttons such as start, step or stop buttons as well as the time (tick) count;



Tick Count: 1366.0

Simulation Parameters

Default Random Seed: 793.370.755

initial number of cars: 12

modeID: 1

xdim: 20

ydim: 20

Paused

Figure 5: Traffic simulator in Repast.

left-side area allows the definition of the setup parameters; middle area shows the actual car simulation; and right-side area is devoted to monitor the evolution of this simulation. Thanks to the setup parameters it is possible to customize current simulation characteristics such as the environment grid dimensions; the maximum number of cars to be simultaneously interacting in the environment; or the learning modality (whose values are 0 if no learning is applied, 1 if a global scope is used in the learning process, and 2 if partial scope). With regards to the actual simulation, cars are represented as circles traversing the two intersecting roads. When cars collide they change their colour to red and disappear. Additionally, a square surrounding a car means that a stop traffic rule has been applied in this specific car position —in the figure example, this specific rule prevents the corresponding car from colliding with the car in front of it. Finally, simulation monitoring shows statistical data about those data that can be useful to follow the evolution of the specified simulation mode. Thus, since the screenshot in figure 5 corresponds to the global scope simulation mode, then the statistical data corresponds to: the number of collisions accumulated during a specific time (tick) window (2000 in the figure); how many stopping rules have been applied for this same period; the total number of cases in the knowledge base; and how many solutions have been explored for this amount of cases.

5.1 Test design

In addition to the development of the simulator it was necessary to conduct a series of experiments in order to evaluate the learning approach. In fact, these experiments were designed sequentially, guided by the results and intuitions gained from previous tests. Our main objective was not to perform an exhaustive search of all possible parameters in the setup process, but a preliminary exploration that gave us some insights about our learning approach. The specific process that we followed can be described in different steps (that are summarized here and detailed in next subsection).

Obviously, we started with the basic simulation mode, in order to asses that cars behave as expected: they drive properly but, since they lack intersection traffic regulations, collisions in the junction area occur with a significant frequency.

Afterwards, we tested the global scope simulation mode. In this case, as next subsection details, we were not able to avoid collisions completely even after running tests for long periods of time (ticks). This was in part due to the limited exploration capacity but also due to the fact that, given the size of the state space, some rare cases actually happen very scarcely, and so, the system did not have the opportunity to explore enough different solutions. This may not invalidate the global approach for all possible scenarios, but it will certainly limit its performance for those domains with large search spaces.

This led us to try the partial approach with the aim of reducing the search space despite its non-determinism problem. Results there were much more promising, since the system was able to find traffic regulations that generated almost no collisions. In addition, it was able to learn them in much shorter periods of time.

Then, by analyzing the resulting regulations, we got the intuition that they could still be described in a shorter way, and thus, we set up a final experiment with cases described by using the minimum amount of information possible.

5.2 Results

Tests with the global scope were performed along a time interval of three million ticks. During this time, the system had the opportunity to visit the whole state space —or, in other words, all possible situations were reached. Nevertheless, after this long period, some collisions –about 10 collisions in a 20000-tick period– still occurred, so the learning process failed to find the proper set of traffic rules that prevented cars from colliding. The reason is two-fold. Firstly, because, despite having encountered all possible cases, more than 20% of the cases remained open (here, cases were closed after exploring five different possible solutions). In fact, from those open cases, almost 80% just had one or two explored solutions. This means that their traffic situations might occur every 1.5 million or more steps on average, and thus, they correspond to what we refer to as rare cases. Therefore, since they happen very scarcely, the system did not had the opportunity to explore enough different solutions so to learn the best ones. The remaining 80% cases did properly close, and therefore, they were finally assigned a single solution —which corresponds to the one with higher performance score[6]. This leads to the second reason, which is the limited exploration capacity over the set of possible solutions. By analyzing the performance of solutions in closed cases, we could observe that, those cases with two or three involved cars were properly regulated, whereas having four cars in the junction lead to some cases –about 10%– whose chosen solution still generated some problems in the traffic flow. Obviously, having a limited number of chances to explore all possible combinations of traffic rules that can be assigned does not guarantee that the best solution will be found. One may argue that this limit should thus be increased, but it would extend the learning time, where collisions can be generated when applying pseudo-random traffic rules.

Having encountered some limitations with the global approach, a second set of experiments with a partial scope were set-up. The main rationale behind this decision was to reduce the size of the search space despite its intrinsic non-determinism problem. The scope was initially defined to be

[6]From the closed cases, just around 20% were in fact problematic in the sense that required the addition of some stopping rules, the rest corresponded to fluid traffic situations.

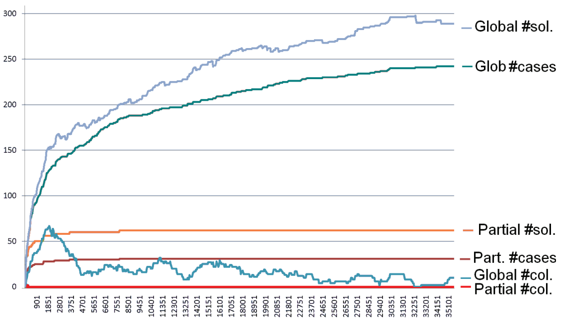


Figure 6: Runtime comparison of global and partial scope learning approaches in terms of the number of: solutions (♯sol.), cases (♯cases), and collisions (♯col.).

a 3×1 grid (as in Figure 4), so each car was able to see a range of 3 cells wide in front of him. Figure 6 plots a comparison between global and partial scopes along first 35100 simulation steps. This comparison is performed in terms of three different measures: the number of generated cases (♯cases); the total number of solutions associated to them (♯sol.); and the number of collisions that occurred during a time window of last 2000 ticks (♯col.). As we can see, learning in partial scope is much faster, since the number of cases stabilizes around 30 much before than the global scope, which tends to the 250 cases along the whole time period that is plotted. Having this small number of cases does not affect the number of collisions. On the contrary, they become zero during initial time steps, which is never the case for the global scope approach (see Figure 6).[7]. Obviously, the whole state space was explored [8] and no case could be considered to be rare. Furthermore, since all cases can just have two possible solutions, we do not consider them to be open or closed —although all of them could somehow be considered to be closed. In addition to avoiding collisions, we were interested in analyzing the kind of solutions that were found. This is so because a formal translation of an automatically learned case solution into a standard norm specification may be of great interest for many MAS. Thus we analyzed those traffic situations that had an stopping regulation and observed that most grids had in common that the cell located in the front left side of the car position was occupied by another car heading (relatively) eastwards —that is, in the direction of the cell the reference car is steering towards. Thus, we can conclude that the system had established a ”left handside priority” traffic rule. And it was so despite the fact that cars were circulating on their right: in real world traffic systems, driving on one side of the road usually comes along with a priority to other participants approaching from the very same side. Tests were repeated in order to find out if the ”right handside priority” was generated. Nevertheless, it was not the case, because the regulations that do not block those cars coming from

[7]CBR learning depends on the order cases are learned. In our case this changes for each new simulation, since the random component on car entrance and route selection may generate traffic situations in different order.

[8]The proportion of problematic cases was very close to the global scope approach.

the left, are in fact promoting a fluid traffic flow within the junction area –similar to roundabout priorities– and thus they got better performances than right handside priorities.

Finally, we wanted to further test if the left handside priority rule was enough to avoid collisions in our traffic simulations. Thus, the last test we did was to repeat partial scope experiments with the minimum range of sight for the reference car: a single cell, the one on its left. Obtained results were really satisfactory, since both the convergence time and the number of collisions was further reduced. From these results, it is possible to argue that the case description in this setting induces the generation of the norm in a straightforward manner, so defining the proper case description may be the underlying problem. Therefore, we do not interpret the positive results obtained with this configuration as the final take-away message. On the contrary, we want to use them as a way that illustrates that learning methods can be used to generate new regulations and that, going a step further, these resulting regulations can be simple enough to be translated into standard traffic rules that can be easily interpreted and followed by external car agents.

6. CONCLUSIONS AND FUTURE WORK

This paper proposes a method to generate new regulations –norms– for multi-agent systems. Specifically, a regulatory authority learns by considering (and exploring) the ones with best application outcome. Learning is based on previous experiences, and corresponds to an unsupervised variation of Case Based Reasoning (CBR). Cases, as defined here, can then be translated to norms, in terms of prohibitions and obligations. We thus claim that this innovative approach can be highly relevant for normative MASs, since, to the best of our knowledge, no general norm generation methods have been established yet.

The paper successfully tests this approach in a simplified traffic scenario. Nevertheless, other scenarios requiring agent coordination –e.g. P2P networks, Robosoccer, etc.– may well benefit from our approach by avoiding (prohibiting) undesired situations –such as network saturation or teammate blocking in previous examples– and promoting (obliging) desired ones. The only requirements[9] are to have monitoring (and evaluating) capabilities as well as continuity in the solution space —i.e., similar social situations require similar regulations. Nevertheless, some undesired situations may appear (e.g, car collisions) as a combination of allowed individual agent actions (e.g., forward driving), thus, norms are required to be more complex than just prohibiting those actions. Context thus becomes necessary. Context, together with its analogy in real Jurisprudence, are the basic rationale of choosing a case representation approach. Nevertheless, we may consider as as future work the application of other learning techniques that cover domains with alternative characterizations. Additionally, we plan to work on norm violation and norm translation issues.

7. REFERENCES

[1] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Commun.*, 7(1):39–59, 1994.

[2] A. Artikis, D. Kaponis, and J. Pitt. *Multi-Agent Systems: Semantics and Dynamics of Organisational Models*, chapter Dynamic Specifications of Norm-Governed Systems. 2009.

[3] G. Boella and L. van der Torre. Regulative and constitutive norms in normative multiagent systems. *Proceedings of KR'04*, pages 255–265, 2004.

[4] L. Busoniu, R. Babuska, and B. de Schutter. A comprehensive survey of multiagent reinforcement learning. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(2):156–172, 2008.

[5] J. Campos, M. López-Sánchez, and M. Esteva. Multi-Agent System adaptation in a Peer-to-Peer scenario. In *ACM Symposium on Applied Computing - Agreement Technologies Track*, pages 735–739, 2009.

[6] G. Christelis and M. Rovatsos. Automated norm synthesis in an agent-based planning enviroment. In *Autonomous Agents and Multiagent Systems (AAMAS)*, pages 161–168, 2009.

[7] N. A. Davis. *Contemporary deontology.* Singer P. (ed) A companion to ethics (Blackwell), 205-218, 1993.

[8] J. Dunkel, A. Fernandez, R. Ortiz, and S. Ossowski. Event-driven architecture for decision support in traffic management systems. In *IEEE Intelligent Transportation Systems Conf.*, pages 7–13, 2008.

[9] J. F. Koeppen. *Norm Generation in Multi-Agent Systems (master thesis).* Univ. of Barcelona, 2009.

[10] R. Kota, N. Gibbins, and N. Jennings. Decentralised structural adaptation in agent organisations. AAMAS Workshop Organised Adaptation in MAS, 2008.

[11] T. McCarty. *Reflections on Taxman: An Experiment in Artificial Intelligence and Legal Reasoning.* Harvard Law Review 837–93, 1977.

[12] J. S. Mill. *Utilitarianism.* Parker, Son, and Bourn (London), 1863.

[13] S. Modgil, N. Faci, F. Meneguzzi, N. Oren, S. Miles, and M. Luck. A framework for monitoring agent-based normative systems. In *Autonomous Agents and Multiagent Systems (AAMAS)*, pages 153–160, 2009.

[14] M. North, T. Howe, N. Collier, and J. Vos. Repast Simphony Runtime System. In *Agent Conf. Generative Social Processes, Models, and Mechanisms*, 2005.

[15] J. H. Powell, B. Hauff, and J. D. Hastings. Evaluating the effectiveness of exploration and accumulated experience in automatic case elicitation. In *Case-Based Reasoning Research and Development, LNAI 3620*, 2005.

[16] B. Savarimuthu, S. Cranefield, M. Purvis, and M. Purvis. Role model based mechanism for norm emergence in artificial agent societies. *Lecture Notes in Computer Science*, 4870:203, 2008.

[17] Y. Shoham and M. Tennenholtz. On social laws for artificial agent societies: off-line design. *Journal of Artificial Intelligence*, 73(1-2):231–252, February 1995.

[18] W. van der Hoek, M. Roberts, and M. Wooldridge. Social laws in alternating time: Effectiveness, feasibility, and synthesis. *Synthese*, 1:156, 2007.

[19] C. Zhang, S. Abdallah, and V. Lesser. Integrating organizational control into multi-agent learning. In *Aut. Agents and Multiagent Systems, 757-764*, 2009.

[9]Obviously, the domain has to be discretisable and a learning phase –where some undesired situations may occur– must be acceptable in that domain. Otherwise, as for the traffic scenario, running simulations may be most adequate.