Using Experience to Generate New Regulations

Javier Morales[1,2] and **Maite López-Sánchez**[1]
[1]MAiA Department
Universitat de Barcelona
Barcelona, Spain
{jmoralesmat, maite_lopez}@ub.edu

Marc Esteva[1]
[2]Artificial Intelligence Research Institute (IIIA)
Spanish Council of Scientific Research (CSIC)
Campus UAB. Bellaterra, Spain
{jmorales, marc}@iiia.csic.es

Abstract

Humans have developed jurisprudence as a mechanism to solve conflictive situations by using past experiences. Following this principle, we propose an approach to enhance a multi-agent system by adding an authority which is able to generate new regulations whenever conflicts arise. Regulations are generated by learning from previous similar situations, using a machine learning technique (based on Case-Based Reasoning) that solves new problems using previous experiences. This approach requires: to be able to gather and evaluate experiences; and to be described in such a way that similar social situations require similar regulations. As a scenario to evaluate our proposal, we use a simplified version of a traffic scenario, where agents are traveling cars. Our goals are to avoid collisions between cars and to avoid heavy traffic. These situations, when happen, lead to the synthesis of new regulations. At each simulation step, applicable regulations are evaluated in terms of their effectiveness and necessity. Overtime the system generates a set of regulations that, if followed, improve system performance (i.e. goal achievement).

1 Introduction

Both human and multi-agent societies have been proven to better function with the inclusion of regulations. In any society, where several individuals continuously interact, conflicts raise naturally. For this reason, within juridical contexts, human societies deal with these conflictive situations by establishing laws that regulate individuals' behavior. Moreover, humans 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 tries to answer questions such as *"What sorts of acts should be punished?"*. In the Anglo-American juridical system, when a conflict arises it is usual to gather information about similar cases used in the past and use their solutions as a reference to solve the current situation. Moreover, when humans solve a new problem, sometimes they generate regulations in order to avoid that problem in the future. Like in human societies, it is possible to enhance the running of a Multi-Agent System (MAS) society by defining specific regulations that promote a desired overall system's behavior. However, key questions are: *"What is the good set of regulations that promote a certain global behavior of the society?"*, and *"How and when to generate new regulations?"*. 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 whenever new conflicts arise.

We approach this regulation generation problem by learning from the experience of MAS on-going activities. As a learning technique to use, we propose a variation of classical Case-Based Reasoning (CBR) [Aamodt and Plaza, 1994]. Classical CBR is a supervised machine learning technique that solves new problems –i.e., cases– by obtaining similar ones from a knowledge base (which is a compound of solved problems) and adapting their solutions under the supervision of an expert. Nevertheless, our aim is to generate proper regulations without external knowledge. Thus, we propose an unsupervised CBR with an initially empty case base that does not require an expert to evaluate generated solutions.

In our approach, a regulatory authority is continuously observing the scenario where agents interact. Then, whenever it perceives a new conflictive situation, it sends its description to our CBR system, which generates a new solution by using previous similar cases. Afterwards, generated case solutions are translated into norms that can be understood by agents (*top-down*). When norms are applicable, agents decide whether to *apply* or *violate* them. Additionally, the regulatory authority continuously observes the outcome of norm applications and violations. This can be regarded as a feedback (*bottom-up*) that evaluates norms in terms of their *effectiveness* and *necessity* according to system goals.

2 Related work

Artificial Intelligence and Law have been related since a first article from McCarty [McCarty, 1977]. Within the MAS area Multi-Agent Reinforcement Leaning [Busoniu *et al.*, 2008] 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.[Zhang *et al.*, 2009] devoted to improve system's orga-

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

nization. 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 MAS is quite an active area. Boella and van der Torre have done relevant contributions [Boella and van der Torre, 2004] in norm characterization; Campos et al. [Campos *et al.*, 2009] have proposed norm adaptation methods to specific network scenarios; Artikis et al.[Artikis *et al.*, 2009] have studied the definition of dynamic social conventions (protocols); and Savarimuthu et al. [Savarimuthu *et al.*, 2008], Griffiths and Luck [Griffiths and Luck, 2010], as well as Kota. et al. [Kota *et al.*, 2008] work on norm emergence. Within this area, norm generation has been studied less frequently. Shoham and Tennenholtz [Shoham and Tennenholtz, 1995] 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 to 3-SAT. Similarly, Hoek et al. [van der Hoek *et al.*, 2007] synthesize social laws as a model checking problem –again NP-Complete– that requires a complete action-based alternative transition system representation. Following this work, Agotnes and Wooldridge [Agotnes and Wooldridge, 2010] extend the model by taking into account both the implementation costs of social laws and multiple (possibly conflicting) design objectives with different priorities. In this setting, the design of social laws becomes an optimization problem. On the contrary, our aim is not to explore the complete search space but just to consider encountered conflictive states, which for most scenarios represent a small proportion of the entire state space. Moreover, CBR has the advantage that, although cases are meant to cover the entire 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 generates norms at run-time. This has the additional advantage of being able to regulate situations that may not be foreseeable at design-time.

An intermediate approach is this of Christelis et al. [Christelis and Rovatsos, 2009; Christelis *et al.*, 2010], that synthesizes generalized norms over general state specifications in planning domains. These domains allow for a local search around declarative specifications of states using AI planning 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 the traffic scenario, we highlight the MAS approach in [Dresner and Stone, 2008], where an intersection agent assigns priorities to traveling cars according to predesigned policies. They follow a control approach that implies a much tighter agent coordination than the one induced in our regulative approach.

3 The traffic scenario

To evaluate our method, we use a simple traffic scenario [Koeppen and López-Sánchez, 2010] which has been developed as a multi-agent based simulation model in Repast [North *et al.*, 2005]. This scenario represents an orthogonal two-road intersection discretized in a square grid of 20×20 cells (see Fig.1). It is divided into five disjoint areas (see Fig.1.b) covered by *monitor agents*. Cars are external agents with basic driving skills that enter into the scenario from four possible start points (dark red points in Fig.1.a), and travel towards randomly chosen destinations (exit points, depicted in light green in Fig.1.a). Time, measured in ticks, is discrete, and cars have a constant *speed* of 1 cell per tick. Cars perform a single action per tick: *move forward*, *stop*, *turn left/right*.

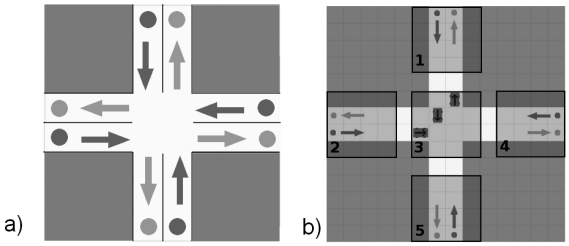

Figure 1: Orthogonal road junction: a) feeder and exit lines, b) traveling cars and areas covered by monitor agents.

4 Norm generation

We enhance our traffic MAS with a regulatory (traffic) authority that generates norms preventing conflictive situations. Inspired in jurisprudence, the authority uses previous experiences (i.e., cases) to avoid new conflicts to arise again in the future. As depicted in Figure 2, the *traffic authority* is permanently observing and gathering information from the scenario through the *monitor agents*. When a new conflict arises the authority sends to the *CBR system* information about its previous and current situation. Then, given the new problem obtained from the *traffic authority*, the CBR looks into the case base for the most similar solved problem and adapts its solution to solve the new one. Since the *traffic authority* initially lacks experience, we require an unsupervised method, and thus, we have adapted CBR to deal with an initially empty case base. For new cases with no previous experience, the system generates random solutions (see section 4.2) and evaluates their performance experimentally. Cases are described from the point of view of *monitor agents*, which have a wider view than car agents. Furthermore, car agents may not be familiar with case syntax, so they are not able to interpret case solutions. Hence, generated case solutions are sent to the *Norms manager*, who translates them to *norms* (which are similar to traffic rules) and establishes permanent links between them and their associated case solutions. Then, generated norms are communicated to all cars, which use a *rule engine* to interpret them. At each step cars use their rule engine to know which norms apply to the current situation, although they decide whether to follow or violate them. Afterwards, the *traffic authority* gathers information about norm applications and violations so to evaluate norms and their associated case solutions in terms of a $score$ (see section 4.4).

4.1 Cases and solutions

In classical CBR cases are described as problems with their associated solutions ($Case = \langle probl, sol \rangle$). Since our approach is unsupervised, for each problem we need to evalu-

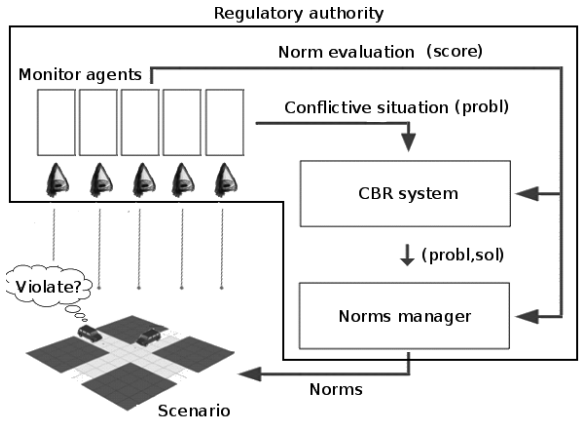

Figure 2: Architecture of our system

ate a set of possible solutions $\{(sol_i, score_i)\}$ so that each sol_i has a $score_i$ value in $[0..1]$. The problem is described as $probl = \langle probl_{t-1}, probl_t \rangle$, where $probl_{t-1}$ (see Fig. 3.a) is the situation previous to the conflict and $probl_t$ (see Fig. 3.b) is the conflictive situation (for example the collision between cars 1 and 2 in Fig. 3.b). Both of them are square grids of 6×6 cells that correspond to the scope of a monitor agent, where each cell contains n cars (being $n >= 0$, and if $n > 1$ means that the cell contains a collision). For each car, the case stores information about its *heading* (North, South, East or West) and whether it is moving or not. The case solution, sol (see Fig. 3.c), is also a square grid with the same dimensions specifying stop obligations to be applied by cars in $probl_{t-1}$.

4.2 Unsupervised CBR cycle

The CBR cycle is composed of 4 different phases: *retrieve, reuse, revise and retain*.

Retrieve phase: Given a new problem the system searches for cases in the case base that have a similar problem. The *similarity* between two cases A and B is computed as the inverse of the *distance* between their case descriptions. It is computed as the aggregation of distances of their attributes, that in our scenario correspond to the cells of $probl$. Specifically, each cell in case A ($c_i^A \in probl_A$) is compared with the corresponding cell in case B ($c_i^B \in probl_B$):

$$dist(probl_A, probl_B) = \sum_{i=1}^{nCells} dist(c_i^A, c_i^B)$$

Differences between two cells are considered to be 1 if their occupancy state is different (notice that this similarity function is commonly used for nominal attributes):

$$dist(c_i^A, c_i^B) = 1 \text{ if } state(c_i^A) \neq state(c_i^B), \text{ where}$$
$$state(c_i^k) = \{empty, car(heading, moving), collision\}$$

Since we may encounter symmetric cases, the retrieve phase searches for rotated versions of the problem:

$$probl_A^\alpha = rotation(probl_A, \alpha), \alpha \in \{0°, 90°, 180°, 270°\}$$

Currently, our implementation of this retrieve phase just returns a single case at most.

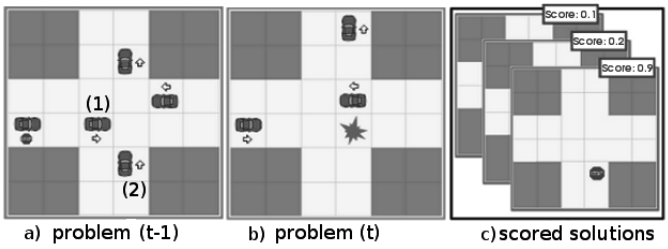

Figure 3: Case example: Problem description (*a*, *b*), being *a)* $probl_{t-1}$, *b)* $probl_t$, and *c)* the set of generated solutions.

Reuse phase: If a case has been retrieved in the previous phase, this phase adapts its best solution to solve the new problem, if the score of this solution is over a *threshold* (see section 4.4). The adaptation process is done by rotating the solution the same α degrees than the problem of the case that was rotated in previous phase to match the new problem. Otherwise, if the $score$ of its best solution is under the *threshold*, the solution is not adapted but an alternative one is generated, assigning a stop obligation to another random car. Then, this new solution is added to the retrieved case. If no similar case was retrieved, a new case, composed by the new problem and a randomly generated solution for it, is created.

Revise phase: In our unsupervised CBR, the revise phase empirically evaluates norms and their associated solutions in an iterative manner (see section 4.4).

Retain phase: This phase stores the resulting experience in the case base. In our unsupervised CBR scenario this may lead to two different possibilities: a) If a new case was generated, it is stored in the case base; b) If an existing case was retrieved, the retain becomes an update of the current case if a new solution was generated or the score of an existing solution was updated.

4.3 From case solutions to norms

Once CBR has generated a case solution, it is sent to the *Norms manager*, who translates it to a norm that car agents can understand and apply. Norms are described as "IF *cond* THEN *obl(action)*", where *cond* is the condition for the norm to be applicable and *obl(action)* is the action that must be performed. Notice that, unless norms only specify obligations (*obl(action)*), they can be also regarded as prohibitions (*prohib(¬ action)*), and that *permission(move)* is granted by default if no norms are applicable. A norm also has a $score$ that is initially set to 0 and represents its performance. Recall that in our traffic scenario a case solution establishes which car has to stop. The content of the 3 cells in front of this car (which corresponds to its scope) is used as the norm condition. The consequence of the norm is the obligation for that car to stop. Figure 4.c depicts the resulting norm from the case example in Fig. 3: top part shows its graphical representation and bottom part its textual form actually used.

4.4 Norm evaluation

Norm evaluation is performed to compute the *effectiveness* and the *necessity* of norms with respect to system goals. Our aim is to accomplish the system goals with the minimum set

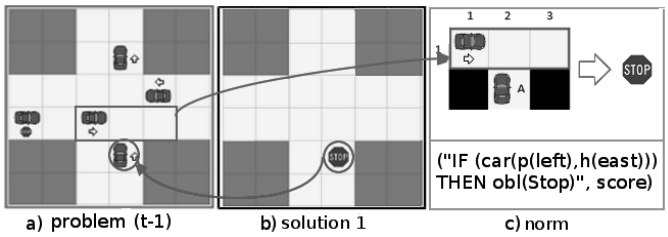

Figure 4: Norm translation from case solution in Fig. 3

of norms, discarding *ineffective* or *unnecessary* norms from our set. On the one hand, the *effectiveness* of a norm is computed taking into account the result of its applications. Specifically, after agents apply the norm it is checked whether a conflict arises (*ineffective norm*) or not (*effective norm*). On the other hand, the *necessity* of a norm is evaluated according to the result of its violations. In particular, after agents have violated a norm it is checked whether a conflict arises (*necessary norm*) or not (*unnecessary norm*). Thus, the time to detect unnecessary norms is inversely proportional to the number of violations. In complex situations, where several norms could be applied (and violated) before a conflict arises, the system would not be not able to discern between norms, and so it just would evaluate all of them equally. Nevertheless, since evaluation is performed in an iterative manner, the method is quite robust to sporadic erroneous evaluations.

Norms are evaluated at each tick considering these four dimensions: *effective, ineffective, necessary, unnecessary*. Specifically, the value for each dimension is calculated by multiplying the number of occurrences of that kind by a constant factor, which should be regarded as the importance given to that kind of situations. Therefore, norms are evaluated using the following formula:

$$\begin{aligned} eval &= effective - ineffective + necessary - unnecessary \\ &= K_E \times Ap_E - K_{\neg E} \times Ap_{\neg E} + K_N \times Viol_N - K_{\neg N} \times Viol_{\neg N} \end{aligned}$$

where Ks are domain-dependent constants that are established by the designer taking into account conflicts and the relative importance of each goal. They define the importance of effective applications (K_E), ineffective applications ($K_{\neg E}$), violations that lead to a conflict (K_N), and violations that lead to a non-conflictive situation ($K_{\neg N}$), Ap_E is the number of norm effective applications, $Ap_{\neg E}$ the number of ineffective applications, $Viol_N$ denotes the number of times a conflict arose after a norm violation, and $Viol_{\neg N}$ the number of times a violation did not lead to a conflict.

Our current scenario considers two different goals, $G = \langle G_{cols}, G_{flTraff} \rangle$. First goal ($G_{cols}$) is to avoid car collisions and second goal is to have fluid traffic ($G_{flTraff}$). Optimizing G_{cols} implies that cars will be occasionally required to reduce speed or to stop in order to avoid collisions. This causes heavier traffic, having an adverse effect on $G_{flTraff}$. On the other hand, $G_{flTraff}$ prefers that cars never stop, which is bad for G_{cols}. If both goals were independent, they could be evaluated separately. However, they are directly related and have conflicting interests, so they have to be evaluated together in order to reach a trade-off between them. Since *ineffective* norms may cause collisions, the *effectiveness* of norms is directly related to the optimization of G_{cols}. *Unnecessary* norms cause unneeded stops and so heavier traffic, being prejudicial for $G_{flTraff}$. Hence, the optimization of $G_{flTraff}$ is performed by avoiding *unnecessary* norms. We can therefore instantiate the evaluation formula as:

$$\begin{aligned} eval = (K_E \times nCAppNoCol) - (K_{\neg E} \times nCAppCol) + \\ (K_N \times nCViolCol) - (K_{\neg N} \times nCViolNoCol) \end{aligned}$$

where $nCAppNoCol$ is the number of cars that applied the norm and did not collide, $nCAppCol$ is the number of cars that applied the norm and collided, $nCViolCol$ is the number of cars that violated the norm and collided, and $nCViolNoCol$ is the number of cars that violated the norm and did not collide. $K_{\neg N}$ is here related to $G_{flTraff}$ because unnecessary norms cause heavier traffic.

Once $eval$ is computed, it is added to the history of evaluations of the norm, which comes down to be a window with $size = sz_{win}$. Finally, the score of the norm is computed by:

$$score = \frac{posEvals}{|negEvals| + posEvals}$$

where $posEvals$ is computed by adding all the values $eval >= 0$ of the evaluation history, and $negEvals$ is computed by adding all the negative evaluation values ($eval < 0$) of the norm. Notice that with this method, the norm is evaluated in an iterative manner. In case the $score$ value becomes under a certain $threshold$, the norm is deactivated and removed from the set of norms. Thus, it will not be applied any longer, unless it is generated again in another conflictive situation. In fact, in order to avoid premature norm deactivations, the system only deactivates norms that have been evaluated a minimum number of times ($minEvals$).

5 Experiments

We have empirically evaluated our approach in a simulation of the traffic scenario described in section 3. Due to the intrinsic randomness of the simulation, each experiment has been repeated 100 different times. Each simulation lasts 10000 $ticks$, and every 2 ticks, 3 new cars are added to the scenario. Thus, during simulations, the number of traveling cars can vary from 23 to 27. When norms are applicable, car agents have a probability $P(Violate) = 0.3$ of violating them. Norms are deactivated when their score is under a $threshold = 0.3$ and they have been evaluated a minimum of 10 times ($minEvals = 10$). Goals of our scenario (G_{cols} and $G_{flTraff}$) are dependent and conflicting. Hence, we have designed two different experiments to study to what extent they can be simultaneously accomplished.

Since their consequences are more dramatic, we consider avoiding collisions to be more important than having fluid traffic. Thus, first experiment just considers G_{cols} to test if the system is able to accomplish it when no other factors are taken into account. For this aim, in this experiment constants for norm efficiency are $K_E = 1$ and $K_{\neg E} = 5$ and constants for norm necessity are $K_N = K_{\neg N} = 0$ (see section 4.4). Thus, the highest punishment is associated to collisions

($K_{\neg E}$). Second experiment considers both goals, although collisions are still considered to be of higher importance. Specifically, in this experiment constants for norm efficiency are $K_E = 1$ and $K_{\neg E} = 5$ and constants for norm necessity are $K_N = 1$ and $K_{\neg N} = 2$. Figure 5 depicts the results of both first (dark/blue thin series) and second (light/green thick series) experiments. All series correspond to the average of 100 simulations. For each tick, top series represent the average number of active norms, middle-high series show the average number of car stops, and bottom series represent the average number of collisions averaged again within a time window of last 50 ticks. Since we evaluate norms, we just consider those collisions caused when norms are applied (instead of also including collisions coming from norm violations). The average of collisions is inversely proportional to the accomplishment of G_{cols}. Similarly, the performance of $G_{flTraff}$ is inversely proportional to the number of car stops. Therefore, these goals can be regarded as the minimization of the number of collisions and car stops respectively.

In experiment 1 the number of stops is always larger than in experiment 2. This is due to the fact that in this experiment any norm that can eventually avoid any collision is included regardless the fact that it may be causing unneeded stops. However, this is taken into account in experiment 2, since the number of stops must also be minimized. On the other hand, in experiment 1 the simulation rapidly converges to a stable set of 10 norms that prevent collisions. Thus, no more collisions happen from $tick$ 550 on although this is at the expense of never optimizing the traffic flow. Experiment 2 has conflicting goals, so the system is continuously activating and deactivating norms to find a trade-off between the performance of G_{cols} and $G_{flTraff}$. Hence, the system does not converge to a stable set of norms. However, resulting norms partially fulfil both goals. Some examples of norms generated in experiments 1 and 2 are the following:

```
1) IF (car(p(left),  h(east)))  THEN obl(Stop)
2) IF (car(p(right), h(west)))  THEN obl(Stop)
3) IF (car(p(front), h(north))) THEN obl(Stop)
4) IF ()                        THEN obl(Stop)
```

Where $p()$ is the *position* of a car and $h()$ is its *heading*. Norms 1 and 2 correspond to the *left-hand side priority* (see Fig. 4) and *right-hand side priority*, respectively. In all performed simulations, preventing the other from appearing. Nevertheless, if both of them are generated, they can lead to deadlocks between two cars approaching from orthogonal directions. Although norm violators will break these deadlocks, these situations will penalize fluid traffic. $G_{flTraff}$ is not considered in experiment 1 but is certainly relevant for experiment 2. Thus, when performing experiment 2, the $score$ of both norms gradually decreases until the system deactivates one of them. From then on, the $score$ of the active norm increases gradually, since it is both *effective* and *necessary*. This allows to accomplish both goals (G_{cols} and $G_{flTraff}$) regardless of the order in which norms 1) and 2) are generated. Notice also that we evaluate norms individually, but since all active norms are tested simultaneously in the scenario, their combination is also evaluated.

Norm 3 can be regarded as a *security distance* norm. It is typically generated and applied in road areas out of the intersection (i.e., areas 1, 2, 4 and 5 in Fig. 1.b). This norm requires the car agent to stop if there is a car heading north in front of him. Since it is preventive, sometimes cars violate it and collide, while sometimes cars violate it and do not collide. It is always active in experiment 1 since it helps to minimize collisions (accomplishment of G_{cols}). Nevertheless, it may seem unnecessary from the point of view of $G_{flTraff}$. In experiment 2 this norm, that goes against one of the goals, is continuously being activated and deactivated because the system is trying to find a trade-off between the optimization of G_{cols} and $G_{flTraff}$. As a consequence, collisions are not completely eradicated, but the number of car stops is reduced with respect to experiment 1. Norm 4 establishes an unconditional stop obligation. It is generated occasionally and rapidly evaluated as *unnecessary* and deactivated.

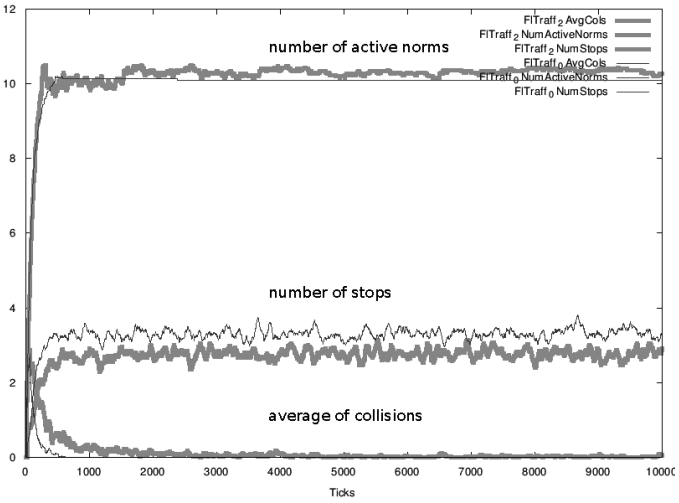

Figure 5: Results for experiments 1 and 2

6 Conclusions

This paper proposes a method to generate new regulations for multi-agent systems. Specifically, regulations are generated by a regulation authority using machine learning, when a conflictive situation arises. Learning is based on previous experiences, and corresponds to an unsupervised variation of Case Based Reasoning (CBR). Cases are expressed in terms of the information observable (accessible) by the regulatory authority, which may differ from the one observable by agents. Hence, generated solutions are translated into norms that are interpretable by agents. Generated norms are evaluated in an iterative manner in terms of their efficiency and necessity according to system goals. The proposed evaluation uses both norm applications and violations. 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.

It is worth mentioning that our aim is to end up with the minimum set of regulations that accomplish system goals. On the one hand, we only generate regulations when a conflictive situation is encountered, which is expected to be a small proportion of the total number of situations a system may face. Hence, this reduces the number of times the CBR

is invoked and the number of generated regulations. On the other hand, the evaluation method helps to discard ineffective and unnecessary norms, reducing the number of norm applications that agents have to check. In addition, our approach supports the indirect evaluation of sets of norms when they apply simultaneously. Although norms are evaluated individually, their evaluation depends on the state reached each time they are applicable, and this state depends on all applicable norms. Thus, if the application of a set of norms leads to a non-conflictive situation, the score of each norm would increase, while if their application leads to a conflictive situation, norms score would decrease.

The paper empirically evaluates our approach in a simplified traffic scenario, where car collisions represent the conflictive situations and norms, that can be regarded as traffic rules, establish under which circumstances a car must stop. Presented experiments illustrate how our approach is capable of generating regulations taking into account a single or multiple conflictive goals. Furthermore, in the second experiment we have shown that when the system generates norms that do not work well together it is capable of discarding some of them. Other scenarios requiring agent coordination –e.g. P2P networks, Robosoccer, etc.– may well benefit from our approach by avoiding conflictive situations —such as network saturation or teammate blocking in previous examples. As future work, we may consider the application of other learning techniques such as Reinforcement Learning. Currently we are working on defining alternative relationships between system goals and metrics for *effectiveness* and *necessity*, measuring both of them for each goal.

Acknowledgements

Work funded by EVE (TIN2009-14702-C02-01 / TIN2009-14702-C02-02) and CONSOLIDER AT (CSD2007-0022) projects and by the Generalitat de Catalunya under the grant 2005-SGR-00093. M. Esteva enjoys a Ramon y Cajal contract from the Spanish Government.

References

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

[Agotnes and Wooldridge, 2010] T. Agotnes and M. Wooldridge. Optimal Social Laws. In *Proceedings of 9th International Conference on Autonomous Agents and Multiagent Systems*, pages 667–674, 2010.

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

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

[Busoniu *et al.*, 2008] 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.

[Campos *et al.*, 2009] 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.

[Christelis and Rovatsos, 2009] 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.

[Christelis *et al.*, 2010] G. Christelis, M. Rovatsos, and R. Petrick. Exploiting Domain Knowledge to Improve Norm Synthesis. In *Proceedings of he Ninth International Conference on Autonomous Agents and Multiagent Systems*, pages 831–838, 2010.

[Dresner and Stone, 2008] K. Dresner and P. Stone. A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence Research*, 31:591–656, March 2008.

[Griffiths and Luck, 2010] N. Griffiths and M. Luck. Norm Emergence in Tag-Based Cooperation. In *9th International Workshop on Coordination, Organization, Institutions and Norms in Multi-Agent Systems. 79-86*, 2010.

[Koeppen and López-Sánchez, 2010] J. Koeppen and M. López-Sánchez. Generating new regulations by learning from experience. In *Proceedings of 9th workshop on Coordination, Organization, Institutions and Norms in Multi-agent Systems*, pages 72–79, May 2010.

[Kota *et al.*, 2008] R. Kota, N. Gibbins, and N. Jennings. Decentralised structural adaptation in agent organisations. pages 54–71. AAMAS Workshop Organised Adaptation in MAS, 2008.

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

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

[Savarimuthu *et al.*, 2008] B.T.R. Savarimuthu, S. Cranefield, M. Purvis, and M. Purvis. Role model based mechanism for norm emergence in artificial agent societies. *Lecture Notes in Computer Science*, 4870:203–217, 2008.

[Shoham and Tennenholtz, 1995] 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.

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

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