

Evaluation of an Automated Mechanism for Generating New Regulations

Javier Morales^{1,2}, Maite López-Sánchez¹, and Marc Esteva²

¹ MAiA Dept., Universitat de Barcelona

{jmoralesmat, maite_lopez}@ub.edu

² Artificial Intelligence Research Institute (IIIA-CSIC)

marc@iiia.csic.es

Abstract. Humans usually use information about previous experiences to solve new problems. Following this principle, we propose an approach to enhance a multi-agent system by including an authority that generates new regulations whenever new conflicts arise. The authority uses a unsupervised version of classical Case-Based Reasoning to learn from previous similar situations and generate regulations that solve the new problem. The scenario used to illustrate and evaluate our proposal is a simulated traffic intersection where agents are traveling cars. A traffic authority observes the scenario and generates new regulations when collisions or heavy traffic are detected. At each simulation step, applicable regulations are evaluated in terms of their effectiveness and necessity in order to generate a set of regulations that, if followed, improve system performance. Empirical evaluation shows that the traffic authority succeeds in avoiding conflicting situations by automatically generating a reduced set of traffic rules.

1 Introduction

In any society, composed by humans or software agents, individuals continuously interact among them, and sometimes conflicts raise naturally. It has been proven that regulations are useful to enhance the turning of societies by regulating individual's behavior and by solving conflictive situations. For instance, within juridical contexts, 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 new conflict arises it is usual to gather information about similar cases that where solved in the past to solve the current problem. Furthermore, when humans solve a new problem, sometimes they generate regulations in order to avoid that problem in the future. MAS societies, like human societies, can be enhanced by including specific regulations that promote a desired system's behavior. However, there are some key questions: "*When to generate new regulations?*", "*How to generate them?*" and "*How to know if the generated set of norms is correct?*".

¹ Jurisprudence definition extracted from Black's Law Dictionary:

<http://www.blackslawdictionary.com>

In a previous work [9] we answered these questions by proposing a computational mechanism that generates norms with the aim to improve the performance of the system. The aim of this paper is to present the resulting norm life cycle that defines the creation, maturing and establishment of sets of norms, as well as a more complete set of experiments.

2 Related Work

Research on norms in MAS is quite an active area. Campos et al. [4] have proposed norm adaptation methods to specific network scenarios; Boella and van der Torre have done relevant contributions [2] in norm characterization. Savarimuthu et al. [10], Griffiths and Luck [6], as well as Kota. et al. [8] work on norm emergence. Within this area, norm generation has been studied less frequently. Shoham and Tennenholtz [11] 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. [7] 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 [1] 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. Our approach does not explore the complete search space. Instead, we just explore a small portion of the search space by just expanding encountered conflictive states. 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. CBR allows the application to a wide 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). Within the MAS area Multi-Agent Reinforcement Learning [3] is quite widely used for individual agent learning. Nevertheless its usage is much more scarce for organizational centered approaches. Regarding Case Elicitation, in [2] an Unsupervised CBR system is used to solve new situations by learning from experience in a checkers game scenario;

Finally, regarding the traffic scenario, we highlight the MAS approach in [5], where an intersection agent assigns priorities to traveling cars according to pre-designed 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

The scenario represents an orthogonal two-road intersection discretized in a square grid of 20×20 cells. It is divided into five (disjoint) adjacent areas (see left of Fig.1) 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 left of Fig.1), and

travel towards randomly chosen destinations (exit points, depicted in light/green in left of Fig.1). Time is discrete (measured in ticks), and cars perform a single *action* $\in \{MoveForward, Stop, TurnLeft, TurnRight\}$ per tick. Cars move at constant *speed* of 1 cell per tick. More details about the scenario can be found in [9].

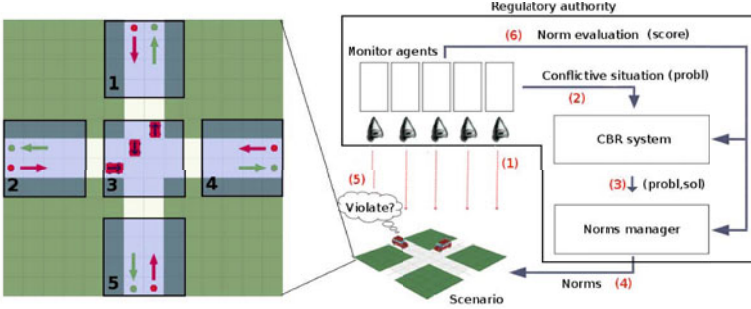


Fig. 1. Left: Zoom of the scenario. Right: Architecture of our system

4 Norm Life Cycle

We enhance our MAS with a regulatory authority (*traffic authority*) that, for new conflicts, generates norms by using the experience of previous similar cases. In the life cycle of a norm there are several stages.

Norm Generation Stage: As depicted in Figure 1, the *traffic authority* is permanently observing and gathering information from the scenario through five *monitor agents* (see label 1 in Fig. 1). When a new conflictive situation is detected, a description of it ($probl = \langle probl_{t-1}, probl_t \rangle$) is sent to the CBR system (2 in Fig. 1), where $probl_{t-1}$ (see Fig. 2.a) is the situation previous to the conflict and $probl_t$ is the conflictive situation (see Fig. 2.b). Then, CBR searches into the case base for cases that have a similar description. Cases are described as $Case = \langle probl, \{sol_i, score_i\} \rangle$, where $probl$ is the case description and $\{sol_i, score_i\}$ corresponds to a list of possible solutions, each one with its associated $score \in [0..1]$. 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 the cells of $probl$, comparing each cell in case A ($c_i^A \in probl_A$) 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, and 0 else (notice that this similarity function is commonly used for nominal attributes):

$$dist(c_i^A, c_i^B) = 1 \text{ if } state(c_i^A) \neq state(c_i^B) \text{ 0 else, where } state(c_i^k) = \{empty, car(heading, moving), collision\}$$

Since we may encounter symmetric cases, CBR applies rotations of α degrees to cases (where $\alpha \in \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$) while retrieving similar problems. When a case has been retrieved the system adapts its best solution to solve the new problem (see Fig. 2.c). The adaptation process is done by rotating the solution the same α degrees than the retrieved case was rotated. Then, this new solution is added to the retrieved case. If the system lacks experience and no similar case was retrieved, a new pseudo-random solution is generated, assigning a stop obligation to one of collided cars.

Car agents may not be familiar with case syntax and so they may not be able to interpret case solutions. Hence, the *Norms Manager* translates case solutions into *norms* that agents can understand (3 in Figure 1). 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 to perform. The norm condition corresponds to the scope of this car, and the consequence of the norm is the obligation for that car to stop. Once a new norm is generated, its *score* is initially set to 0. Figure 2.d depicts the resulting norm from the case example. Top part shows its graphical representation and bottom part its textual form. Generated norms are then communicated to the agents (see label 4 in Figure 1).

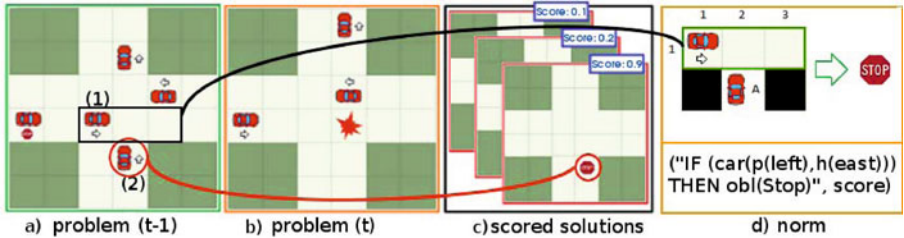


Fig. 2. Norm generation example: Case description in terms of a) $probl_{t-1}$ and b) $probl_t$; c) the set of generated solutions, and the resulting translated norm in d), where $p()$ is the position of the car and $h()$ is its heading

Norm Growth Stage: At each step cars use a *rule engine* to interpret which norms are applicable and decide whether to apply or violate them (5 in Figure 1). The *traffic institution* continuously gathers information about norm applications and violations so to evaluate norms (6 in Figure 1) and their associated case solutions in terms of a *score*. At each tick, the *traffic institution* detects which norms are applicable and evaluates them with respect to to system goals considering their *effectiveness* and *necessity*. Specifically, norm applications are used to compute the *effectiveness* of a norm, checking whether a conflict arises (*ineffective norm*) or not (*effective norm*) after agents apply it. Norm violations are used to evaluate the *necessity* of a norm, checking whether a conflict arises (*necessary norm*) or not (*unnecessary norm*) after agents violate it. Therefore, norms are evaluated using the following formula:

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

where Ap_E/Ap_{-E} are the number of applications that were effective/ineffective, and $Viol_N/Viol_{-N}$ denote the number of times a violation did/did not lead to a conflict.

Specifically, the value for each dimension is calculated by multiplying the number of occurrences of that kind by a constant factor K_i which is established by the designer and should be regarded as the importance given to that kind of situations.

Our current scenario considers two different goals, $G = \langle G_{cols}, G_{flTraff} \rangle$ which are directly related and contradictory. First goal (G_{cols}) is to avoid car collisions and second goal is to have fluid traffic ($G_{flTraff}$). On the one hand, optimizing G_{cols} requires cars to occasionally reduce speed or to stop in order to avoid collisions, causing heavier traffic and decreasing the performance of $G_{flTraff}$. On the other hand, optimizing $G_{flTraff}$ requires car not to stop, which decreases the performance of G_{cols} . Both goals are 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 $G_{flTraff}$. We can therefore instantiate the evaluation formula as:

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

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. 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 $eval \geq 0$ 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.

Norm Consolidation/Deactivation Stage: After norms have been evaluated a minimum number of times ($minEvals$), they are considered to have accumulated experience enough to determine if they must remain active or not. 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. Otherwise, if the norm remains active and its score is stable during the simulation, it is consolidated and considered as part of the optimal set of norms that optimize the running of the system.

5 Experiments

In order to evaluate our method and to compare its efficiency with standard coordination mechanisms (that is, with traffic lights) we have designed 4 different experiments. All experiments have been executed over the same simulator of the traffic scenario described in section 3. 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 $G_{flTraffic}$ 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. Goals of our scenario (G_{cols} and $G_{flTraffic}$) are dependent and conflicting.

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 that simultaneously populate the scenario can vary from 23 to 27. When norms are applicable, car agents have a probability $P(Violate) = 0.3$ of violating them. The size of the evaluations window is $size_{win} = 50$. Norms are deactivated when their score is under a $threshold = 0.3$ and they have been evaluated a minimum of 10 times ($minEvals = 10$) (see section 4).

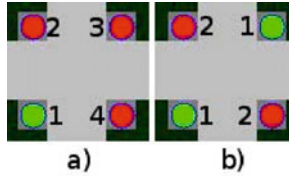


Fig. 3. Different configurations for traffic lights: a) 4 green light turns (1-East, 2-South, 3-West, 4-North) b) 2 green light turns (1-East & West, 2-North & South)

Experiment 1 uses our norm generation method to regulate the intersection. This experiment just considers G_{cols} in order to test if the system is able to accomplish one goal when no other factors are taken into account. For this aim, in this experiment constants are $K_E = 1$, $K_{-E} = 5$ and $K_N = K_{-N} = 0$. In order to compare our method with standard methods established by humans, in experiment 2 the scenario is regulated by traffic lights situated before entering the intersection, and there is no norm generation method. This approach is also used by K. Dressner and P. Stone in [5]. There are 4 lights, one for each lane. Traffic lights change their lights in 4 turns, as depicted in Figure 3.a. Thus, they give pass to the cars of one only lane at the same time.

Figure 4 depicts the results of both experiments 1 and 2. In experiment 1, using our norm generation method, the number of car stops is always lower than in experiment 2 (about 26 car stops per tick with traffic lights, and 4 car stops per tick with our method). This is due to the fact that, with traffic lights, cars are forced to stop following fixed patterns (i.e, time frequencies) regardless the actual traffic situation or traffic flow. On the other hand, with our approach norms describe situations and force cars to stop depending on a finer detail (the position of other cars). Thus, our method obtains a better performance for $G_{flTraffic}$. In experiment 2 the traffic lights configuration totally avoid collisions since traveling cars never find cars from another lane into the intersection. In experiment 1, collisions are eradicated from *tick* 550 on, optimizing the performance of G_{cols} . Since the system has one only goal G_{cols} , all norms that can eventually avoid collisions are included regardless the fact that they may be causing unneeded stops. Thus, the system rapidly converges to a stable set of 10 active norms that prevent collisions.

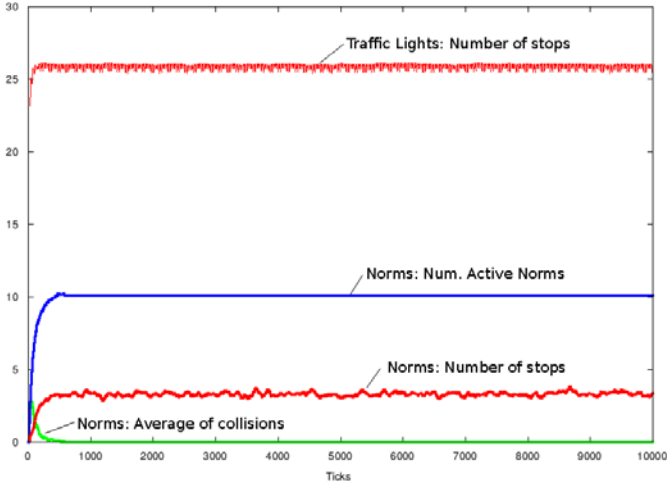


Fig. 4. Comparison of the results of experiments 1 and 2. Experiment 1: Norm generation with the goal of minimizing collisions. Experiment 2: Traffic light configuration of Figure 3.a.

In order to study how norms can be generated when considering multiple conflicting goals, we have conducted a third experiment that applies our norm approach and considers both conflicting goals G_{cols} and $G_{flTraffic}$. Specifically, constants are $K_E = 1$, $K_{\neg E} = 5$ and $K_N = 1$ and $K_{\neg N} = 2$. We compare the results of this experiment 3 with a fourth experiment, where an alternative setting of traffic lights improves the performance of fluid traffic. This is done by giving pass to two lanes simultaneously (see Figure 3.b), with the associated penalty that collisions may happen into the intersection.

As depicted in Figure 5, the number of car stops per tick has decreased ($\simeq 19$) with respect to experiment 2. In experiment 3, using our method, the number of car stops has also decreased since now they are also part of the system goals ($G_{flTraffic}$). Moreover, our method also optimizes fluid traffic much better than traffic lights, while collisions are relatively controlled ($\simeq 0.2$ per tick). In experiment 4, the average of collisions remains always higher than in experiment 3. Hence, our method optimizes both G_{cols} and $G_{flTraffic}$ in a better way than traffic lights. Experiment 3 has conflicting goals, so the system is continuously activating and deactivating norms to find a trade-off between the performance of G_{cols} and $G_{flTraffic}$. Hence, the system does not converge to a stable number of active norms. However, resulting norms partially fulfill both goals.

Two typical norms that always appear in all the performed simulations are:

- 1) IF (car(pos(left), heading(east))) THEN obl(Stop)
- 2) IF (car(pos(front), heading(north))) THEN obl(Stop)

Where $pos()$ is the *position* of a car and $heading()$ is its *heading*. Norm 1 corresponds to the *left-hand side priority* (see Fig. 2). In all performed simulations this norm (or its counterpart, the *right-hand side priority*) is always generated. This norm requires the car agent to stop if there is a car heading east to his left. In experiment 1, that uses

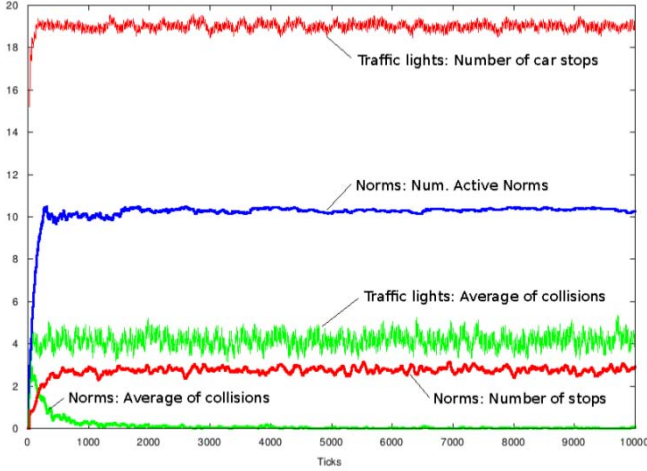


Fig. 5. Results for experiments 3 and 4. Experiment 3: Norm generation with both goals of minimizing collisions and to have fluid traffic. Experiment 4: Traffic light configuration of Figure 3.b.

our norm generation method and takes into account only G_{cols} , this norm is generated and finally consolidated, but never deactivated. Thus, it is always part of the final set of norms. This is due to the fact that this norm is necessary, since its violations always lead to a collision. In experiment 3, that uses our approach and takes into account G_{cols} and $G_{flTraff}$, norm 1 is also always generated and consolidated. Since it is necessary, the norm also contributes to improve the performance of $G_{flTraff}$, being considered as part of the final set of norms that improve the running of the system.

Norm 2 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 the left of Fig. 1). This norm requires the car agent to stop if there is a car in front of him with its same heading. Since it is preventive, sometimes cars violate it and collide, while some other times cars violate it and do not collide. In experiment 1, $G_{flTraff}$ is not taken into account and so this norm, that may seem unnecessary from the point of view of $G_{flTraff}$, is always included regardless the fact that it sometimes may cause unnecessary stops. Then, in this experiment it is generated and finally consolidated since it helps to minimize collisions (accomplishment of G_{cols}). In experiment 3 this norm, that goes against one of the goals ($G_{flTraff}$), is continuously being activated and deactivated because the system is trying to find a trade-off between the optimization of G_{cols} and $G_{flTraff}$. Specifically, this norm is always generated and occasionally it is deactivated because it is unnecessary from the point of view of $G_{flTraff}$. However, since the norm is necessary from the point of view G_{cols} , the norm is later generated again in another case and its life cycle starts again. As a consequence, collisions are not completely eradicated, but the number of car stops is reduced with respect to experiment 1.

6 Conclusions

This paper proposes a method to generate new regulations for multi-agent systems. Specifically, regulations are generated by a regulation authority using an unsupervised variation of Case Based Reasoning (CBR), when a conflictive situation arises. Generated norms are evaluated in an iterative manner in terms of their efficiency and necessity according to system goals. 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 that are capable to adapt the set of regulations during the execution of the system. 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. Applicable norms are then evaluated as a set of norms. 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.

This paper empirically evaluates our approach in the simulation of a simplified traffic scenario, where car collisions and traffic jams represent the conflictive situations and norms establish which circumstances a car must stop. Presented experiments compare our approach with standard traffic regulation methods like traffic lights. Results show how our method is capable to generate effective regulations taking into account single or multiple goals, improving the performance of system goals in a higher level than traffic lights.

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 our approach in other scenarios like these ones that have been just mentioned, and the application of other learning techniques such as Reinforcement Learning.

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

1. Agotnes, T., Wooldridge, M.: Optimal Social Laws. In: Proceedings of 9th International Conference on Autonomous Agents and Multiagent Systems, pp. 667–674 (2010)
2. Boella, G., van der Torre, L.: Regulative and constitutive norms in normative multiagent systems. In: Proceedings of KR 2004, pp. 255–265 (2004)
3. Busoniu, L., Babuska, R., de Schutter, B.: A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 38(2), 156–172 (2008), <http://dx.doi.org/10.1109/TSMCC.2007.913919>
4. Campos, J., López-Sánchez, M., Esteva, M.: Multi-Agent System adaptation in a Peer-to-Peer scenario. In: ACM Symposium on Applied Computing - Agreement Technologies Track, pp. 735–739 (2009)

5. Dresner, K., Stone, P.: A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence Research* 31, 591–656 (2008)
6. Griffiths, N., Luck, M.: Norm Emergence in Tag-Based Cooperation. In: 9th International Workshop on Coordination, Organization, Institutions and Norms in Multi-Agent Systems, pp. 79–86 (2010)
7. van der Hoek, W., Roberts, M., Wooldridge, M.: Social laws in alternating time: Effectiveness, feasibility, and synthesis. *Synthese* 1, 156 (2007)
8. Kota, R., Gibbins, N., Jennings, N.: Decentralised structural adaptation in agent organisations. In: AAMAS Workshop Organised Adaptation in MAS, pp. 54–71 (2008)
9. Morales, J., López-Sánchez, M., Esteva, M.: Using Experience to Generate New Regulations. In: Proceedings of the 22th International Joint Conference on Artificial Intelligence, IJCAI (2011)
10. Savarimuthu, B., Cranefield, S., Purvis, M., Purvis, M.: Role model based mechanism for norm emergence in artificial agent societies. In: Sichman, J.S., Padget, J., Ossowski, S., Noriega, P. (eds.) COIN 2007. LNCS (LNAI), vol. 4870, pp. 203–217. Springer, Heidelberg (2008)
11. Shoham, Y., Tennenholtz, M.: On social laws for artificial agent societies: off-line design. *Journal of Artificial Intelligence* 73(1-2), 231–252 (1995)