Toward a Self-Learning Governance Loop for Competitive Multi-Attribute MAS*

Michael Pernpeintner[0001−6939−1028]

Institute for Enterprise Systems (InES), University of Mannheim, Germany
pernpeintner@es.uni-mannheim.de

Abstract. Competitive Multi-Agent Systems (MAS) are inherently hard to control due to agent autonomy and strategic behavior, which is particularly problematic when there are system-level objectives to be achieved or specific environmental states to be avoided.
Existing methods mostly assume specific knowledge about agent preferences, utilities and strategies, neglecting the fact that actions are not always directly linked to genuine agent preferences, but can also reflect anticipated competitor behavior, be a concession to a superior adversary or simply be intended to mislead other agents. This assumption both reduces applicability to real-world systems and opens room for manipulation.
We therefore propose a new governance approach for Multi-Attribute MAS which relies exclusively on publicly observable actions and transitions, and uses the acquired knowledge to purposefully restrict action spaces, thereby achieving the system's objectives while preserving a high level of autonomy for the agents.

Keywords: Multi-Agent System · Competition · Governance · Restriction.

1 Introduction

One of the most intriguing and challenging characteristics of an MAS is the fact that the environmental transitions depend simultaneously on the actions of all agents. This mutual influence leads to strategic and sometimes even seemingly erratic agent actions—particularly when human agents are involved—, and it decouples *intended* and *observed* behavior: In general, the preference order of a self-interested and strategic agent over the environmental states cannot be concluded from observing its actions, meaning that preference elicitation, for example using CP-nets [4], is only possible as long as additional assumptions hold about the link between actions and preferences.

While full control on the part of an outside authority directly contradicts the Multi-Agent property of such a system, some level of control and cooperation can still be achieved. Existing approaches include Stochastic Games [8], Deontic

* This work is supported by the German Federal Ministry of Transport and Digital Infrastructure (BMVI).

Logic [7], Normative Systems [9], and more specifically Normative Multi-Agent Systems [2, 1] and Game Theory for MAS [5, 3, 6]. Building upon these ideas, we propose to make use of the knowledge collected by observing how agents behave in the system, in order to refine the rules of the game. As a consequence, we do not reason in terms of agent preferences or utilities, but rather in terms of actions and transitions.

Naturally, there is a conflict between control and autonomy, requiring a relative weighting of the two objectives. We strive here for minimal restriction, subject to a constraint on the expected value of the system objective.

2 Model

Consider a finite set $\mathcal{I} = \{1, ..., n\}$ of agents who, at every time step $t \in \mathbb{N}_0$, perceive the environmental state $s_t \in \mathcal{S}$ and perform an action $a_i \in \mathcal{A}_i, i \in \mathcal{I}$, following a confidential *action policy* $\pi_i : \mathcal{S} \to \mathcal{A}_i$. The environmental state changes from t to $t+1$ according to a *transition function* $\delta : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, where $\mathcal{A} = \prod_i \mathcal{A}_i$.

Since the action policies are at the agents' discretion, the evolution $s_{t+1} := \delta\left(s_t, \pi(s_t)\right)$ can be influenced either by changing *what agents can do* (altering their action sets) or by changing *what consequences actions have* (altering the transition function). We choose a strict separation of concerns: δ represents the (unalterable) reaction of the environment to agent actions, while the restriction of actions is performed by the Governance $\mathcal{G}$. To use an analogy, the transition function accounts for the laws of nature, whereas the Governance plays the role of the legislature.

The Governance intervenes by defining a set of *allowed actions* $A_t = \Gamma(s_t) \sqsubseteq \mathcal{A}$, where $\mathcal{A} = \prod_i \mathcal{A}_i$ is the *fundamental action set*. When all agents have made their choice $a = (a_i)_i \in A_t$, the Governance uses the information (s_t, a) to learn, i.e., to update its internal state $s_{\mathcal{G}}^{(t+1)} := \lambda\left(s_{\mathcal{G}}^{(t)}, s_t, a\right)$.

We assume that there is a system objective in addition to the agents' goals. Since $\mathcal{G}$ has only probabilistic information about the agents' future actions, its objective needs to be compatible with probabilistic reasoning. Therefore, we assume a cost function $c_{\mathcal{G}} : \mathcal{S} \to \mathbb{R}$ to be minimized.

3 Governance Loop

In this work, we propose a solution for multivariate binary environments ($\mathcal{S} = \mathbb{B}^m$ for some $m \in \mathbb{N}$), where agents can change one attribute per time step (or choose the *neutral action* $\varnothing$), and an attribute is toggled when at least one agent chooses to change it.

Let n be the number of agents, m the number of attributes, and q the number of actions per agent (we assume the same $\mathcal{A}_i$ for all i).

3.1 Observation and Learning

Let $s_{\mathcal{G}}$ be a simple counter of observed actions per agent per environmental state, i.e., $\mathcal{S}_{\mathcal{G}} := \mathbb{N}_0^{n \cdot 2^m \cdot q}$. For every observation (s_t, a_t), the learning function λ increments the respective numbers by one.

This gives rise to an (observed) probability distribution

$$P_i^{(t)}(s) := \left(\frac{s_{\mathcal{G}}^{(i,s,1)}}{s_{\mathcal{G}}^{(i,s)}}, ..., \frac{s_{\mathcal{G}}^{(i,s,q)}}{s_{\mathcal{G}}^{(i,s)}} \right) \text{ , where } s_{\mathcal{G}}^{(i,s)} = \sum_{k=1}^{q} s_{\mathcal{G}}^{(i,s,k)}$$

for all i and s, reflecting the knowledge about the agents' actions up to step t and thus being $\mathcal{G}$'s best guess for the actions at $(t+1)$. It is customary to set $P_i^{(t)}(s) := \left(\frac{1}{q}, ..., \frac{1}{q}\right)$ if $s_{\mathcal{G}}^{(i,s)} = 0$.

3.2 Restriction of Action Spaces

Given some independence assumptions, Algorithm 1 solves the restriction problem by computing an *expected cost matrix* for all joint actions, and then deleting individual actions from this matrix until the expected value drops below a given cost threshold α.

Data: Governance cost function $c_{\mathcal{G}}$, cost threshold α
Input: Agent-specific probability distributions $P_i(s_t)$
Output: Restricted action set A_t
$P(s_t) := \prod_n P_i(s_t) \in \mathbb{P}_q^n$;
$C := P(s_t) \circ c_{\mathcal{G}} \in \mathbb{R}^{q^n}$;
$A := \mathcal{A}$;
while $\sum_{a \in A} C_a > \alpha$ **do**
  $(i,j) := \arg\max_{a_j \in A_i \setminus \{\varnothing\}, i \in \mathcal{I}} C_{(\square_{-i}, a_j)}$;
  Remove all $a \in A$ where agent i chooses a_j and delete the corresponding hyperplane of C ;
end
$A_t := A$;

Algorithm 1: Restricting agent actions

Theorem 1 (Proof omitted). *Let $\alpha \geq C_{\delta(s_t, \varnothing)}$. Then Algorithm 1 produces a restriction $A_t \subseteq \mathcal{A}$ of actions such that $C_{A_t} \leq \alpha$. This restriction is Pareto minimal, i.e., $\nexists A_t' \sqsupset A_t$ with the same property.*

If the cost function has the form $c_{\mathcal{G}}(s) := \mathbb{1}_{\mathcal{S}_-}(s)$ for a subset $\mathcal{S}_- \in \mathcal{S}$ of *violating states*, then α is precisely an upper bound for the probability of transitioning into a violating state.

The worst-case time complexity of Algorithm 1 is $\mathcal{O}\left(n^2 \cdot q^{n+2}\right)$.

4 Evaluation and Results

We compare unrestricted (agents have the full range of actions) and restricted (with Governance as in Section 3) evolution. To quantify the restriction, we use $\mathfrak{r}_{\mathcal{G}}(t) := 1 - \frac{|A_t|}{|\mathcal{A}|} \in [0,1]$ and show this *degree of restriction* together with the average cost over time.

The application domain is a smart home environment with 7 binary variables: $\mathcal{S} = T \times O \times W \times B \times H \times L \times A \cong \mathbb{B}^7$ (Time, Occupancy, Window, Blinds, Heating, Light and Alarm). Agents can change five of the variables, while time and occupancy are controlled externally. The Governance wants to make sure that the heating is off whenever the window is open, and therefore acts against the cost function $c_{\mathcal{G}}(s) := \mathbb{1}_{s_W \wedge s_H}$.

In the deterministic scenario, each agent i has a fixed mapping $\pi_i : \mathcal{S} \to \mathcal{A}_i$ of states to actions. In the probabilistic scenario, agents have probability distributions for their action policy $\pi_i : \mathcal{S} \to \Delta\mathcal{A}_i$.

Each line in Figure 1 is the mean of 10 independent runs with identical parameters, random initial states and fixed $\alpha := \frac{3}{2} \cdot \frac{1}{q^n}$.

4.1 Results

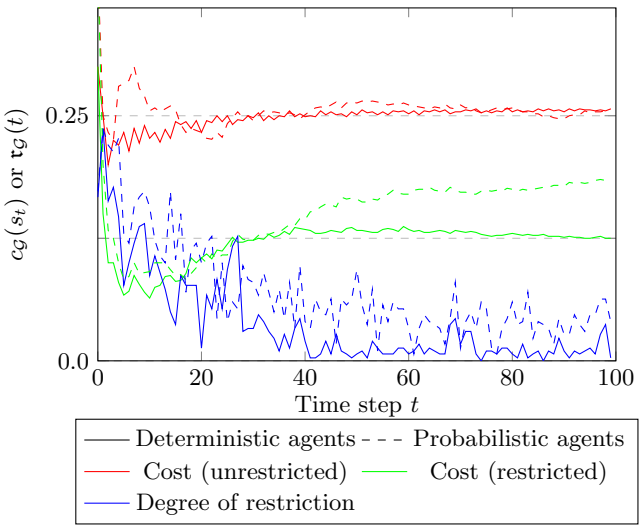

Fig. 1. Evaluation results

$\mathcal{G}$ succeeds in reducing the average cost substantially in all cases from the a priori violation probability of 25%. Moreover, both cost and degree of restriction decrease over time, which indicates that the Governance indeed learns to predict agent actions and improves its corrective action. Notably, this learning process is

independent from an estimated agent preference order: The action policies were created randomly, which means that they most likely do not correspond to a consistent order over the environmental states.

5 Conclusion and Future Work

We show here that governing a competitive MAS is possible without prior knowledge or assumptions about agent preferences. This extends the applicability of such an approach to unknown and, in particular, human agents.

While the algorithm is functional, it lacks (polynomial) scalability in terms of the number of agents and attributes, and it fully re-evaluates the minimal restriction at every step, thereby reducing continuity of allowed actions over time. Future work will therefore include a more efficient representation of knowledge (e.g. attribute dependencies and conditional probabilities), as well as a generalization to environments with continuous attributes or irregular shape, more complex agent actions, locality constraints and multiple-step restrictions.

References

1. Andrighetto, G., Governatori, G., Noriega, P., van der Torre, L.: Normative Multi-Agent Systems (Apr 2013). https://doi.org/10.4230/DFU.Vol4.12111.i
2. Boella, G., van der Torre, L., Verhagen, H.: Introduction to normative multiagent systems. Computational & Mathematical Organization Theory **12**(2), 71–79 (Oct 2006). https://doi.org/10.1007/s10588-006-9537-7, https://doi.org/10.1007/s10588-006-9537-7
3. Brafman, R.I., Tennenholtz, M.: On partially controlled multi-agent systems. J. Artif. Int. Res. **4**(1), 477–507 (Jun 1996)
4. Koriche, F., Zanuttini, B.: Learning conditional preference networks. Artificial Intelligence **174**(11), 685–703 (Jul 2010). https://doi.org/10.1016/j.artint.2010.04.019, http://www.sciencedirect.com/science/article/pii/S000437021000055X
5. Littman, M.L.: Markov games as a framework for multi-agent reinforcement learning. In: Proceedings of the eleventh international conference on international conference on machine learning. pp. 157–163. ICML'94, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1994)
6. Liu, T., Wang, J., Zhang, X., Cheng, D.: Game theoretic control of multiagent systems. SIAM Journal on Control and Optimization **57**, 1691–1709 (Jan 2019)
7. Meyer, J.J.C., Wieringa, R.J. (eds.): Deontic logic in computer science: Normative system specification. John Wiley & Sons, Inc., USA (1994)
8. Shapley, L.S.: Stochastic Games. Proceedings of the National Academy of Sciences of the United States of America **39**(10), 1095–1100 (Oct 1953), https://pubmed.ncbi.nlm.nih.gov/16589380
9. Shoham, Y., Tennenholtz, M.: On social laws for artificial agent societies: off-line design. Artificial Intelligence **73**(1), 231 – 252 (1995). https://doi.org/https://doi.org/10.1016/0004-3702(94)00007-N, http://www.sciencedirect.com/science/article/pii/000437029400007N