# Power Indices for Team Reformation Planning Under Uncertainty

Extended Abstract

Jonathan Cohen
GREYC-CNRS Lab, University of Caen Normandy
Caen, France
jonathan.cohen@unicaen.fr

Abdel-Illah Mouaddib
GREYC-CNRS Lab, University of Caen Normandy
Caen, France
abdel-illah.mouaddib@unicaen.fr

## ABSTRACT

This work is an attempt at solving the problem of decentralized team formation and reformation under uncertainty with partial observability. We describe a model coined Team-POMDP, derived from the standard Dec-POMDP model, and we propose an approach based on the computation of team power indices using the Elo rating system to determine the most fitting team of agents in every situation. We couple this to a Monte-Carlo Tree Search algorithm to efficiently compute joint policies.

## KEYWORDS

Coalition formation; Teamwork, team formation; Multi-agent planning

**ACM Reference Format:**
Jonathan Cohen and Abdel-Illah Mouaddib. 2019. Power Indices for Team Reformation Planning Under Uncertainty. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 3 pages.

## 1 INTRODUCTION

The decentralized POMDP (Dec-POMDP) is the standard model for sequential cooperative multi-agent planning where agents have partial observability of their environment and no way to communicate with each other [3, 9, 13]. Dec-POMDPs consider dynamic, stochastic environments, but do not account for the dynamics of the team of agents. In many real-life scenarios however, it is important to form the right team of agents to solve a task optimally. Moreover, it is also necessary to *dynamically re-form* the team of agents as the task evolves over time. Some related work exists, but the problem of *team reformation planning under uncertainty*, where agents can go in and out of the system on the fly, is novel in itself. The various aspects of *agent openness* in multi-agent systems have only been seldom studied [1, 5, 11, 12, 15–18].

This paper extends the Dec-POMDP model by introducing a model coined *Team-POMDP*, allowing to plan for different teams of agents. Because Dec-POMDPs – and, thus, Team-POMDPs – are computationally hard to solve [3], we need to mitigate the complexity of having to plan for a large number of admissible teams. We notice that, in some states, some compositions of agents are more efficient that others. This observation allows us to prune the policy space by avoiding using bad teams and, hence, bad joint actions.

*Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.* 

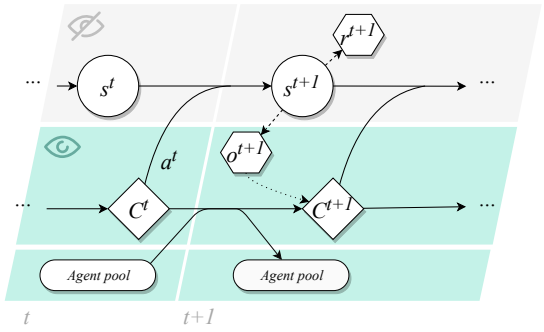


**Figure 1: Depiction of the flow of the Team-POMDP model. A transition from decision epochs $t$ to $t+1$ is represented.**

We use the Elo rating system to provide and update power indices to rank each team of agents. We then wire this rating system to a Monte-Carlo Tree Search inspired planning algorithm to compute joint policies.

## 2 THE TEAM-POMDP MODEL

A *team-formation Partially Observable Markovian Decision Process* (*Team-POMDP*) [6, 7] is a tuple

$$(\mathcal{N}, \mathcal{C}, \mathcal{S}, \{\mathcal{A}_C\}, \{\mathcal{O}_C\}, \{P_{s,s'}^{a_C,o_C}\}, \{R_s^{a_C}\}, b^0, H)$$

where $\mathcal{N}$ is a population of $n$ agents, $\mathcal{C} \subseteq 2^{\mathcal{N}}$ the set of admissible teams, $\mathcal{S}$ the set of states, $\mathcal{A}_C = \times_{i \in C} \mathcal{A}_i$ the set of joint actions of team $C$ and $\mathcal{A}_i$ the set of individual actions of agent $i$, $\mathcal{O}_C = \times_{i \in C} \mathcal{O}_i$ the set of joint observations of team $C$ and $\mathcal{O}_i$ the set of individual observations of agent $i$, $P_{s,s'}^{a_C,o_C}$ is the probability for the system to transit to $s' \in \mathcal{S}$ and emitting observation $o_C \in \mathcal{O}_C$ when team $C$ takes joint action $a_C \in \mathcal{A}_C$ in $s \in S$, $R_s^{a_C}$ is the reward for taking joint action $a_C \in \mathcal{A}_C$ in $s \in S$, $b^0$ is the initial belief state (a probability distribution over $\mathcal{S}$), and $H$ is the planning horizon. Figure 1 depicts how a Team-POMDP unfolds upon execution. The process starts at $t = 0$ with the system in an initial state $s^0$ sampled from $b^0$. At each time step $t$, the system is in a state $s^t$. A team $C^t \in \mathcal{C}$ is selected and chooses a joint action $a^t \in \mathcal{A}_{C^t}$ according to some policy. The system then transits from $s^t$ to $s^{t+1}$. A reward $r^{t+1} = R_{s^{t+1}}^{a^t}$ is generated and a joint observation $o^{t+1} \in \mathcal{O}_{C^t}$ is emitted to the agents of $C^t$. A new team $C^{t+1} \in \mathcal{C}$ (it can be the same as $C^t$) is then selected, with some agents leaving the agent pool, and some other rejoining it. This process repeats for a period of $H$ time steps, until $t = H - 1$.

## 3 POWER INDICES

When considering teams of agents, it is clear that, in some situations, some agents will be more efficient than others to help solving a task. This is due to the fact that an agent possesses its own set of individual actions (that can be unique among the whole pool of agents) and thus its own characteristics and abilities. Even if all the agents are homogeneous (*i.e.* if they all share the same individual actions), using the largest team (composed of the whole population $\mathcal{N}$) can turn out to be counter productive, because agents can interfere with each other or generate more costs than profits. Being able to quickly identify efficient compositions of agents is then of primary importance.

At a certain joint history of observations, determining the most fitting team for the situation can be seen as finding the winner of a competition between all the admissible teams of $C$. The *winning* team at a joint history is the one that will accrue the largest long-term reward when employed at the current joint history. We use the Elo rating system [10] to compute the relative utilities of the teams. Consider two teams $C_1, C_2 \in C$ and a joint history $\vec{o}$. The teams respectively start with initial Elo ratings $E^{\vec{o}}_{C_1} = E^{\vec{o}}_{C_2} \in \mathbb{R}$. Those values are iteratively updated in order to fit the true relative value of the teams. The update equation we use to update a team's rating is the equation traditionally used in official chess competitions:

$$E^{\vec{o}}_{C_1} \leftarrow E^{\vec{o}}_{C_1} + \kappa(W_{C_1,C_2} - L_{C_1,C_2}) \tag{1}$$

where $\kappa$ is a constant, $W_{C_1,C_2}$ is the *actual* result of the match between $C_1$ and $C_2$, and $L_{C_1,C_2}$ is the *expected* result of the match between $C_1$ and $C_2$.

By running a sufficient number of matches against the other teams and updating a team's Elo rating with Equation 1, the system is known to converge to the team's true skill rating.

## 4 MONTE-CARLO PLANNING

Our algorithm is based on the famous Monte-Carlo Tree Search (MCTS) method [4, 8]. This algorithm has proven particularly effective in solving problems and games that can be represented using tree structures.

In the context of a Team-POMDP, we begin with a directed rooted tree, called the *search tree*, made of a single node, the root. A node in the search tree corresponds to a joint history of observations and stores four pieces of information: a *joint observation* $o \in O$, an array of *count numbers*, counting how many times each joint action has been tried by each team, an array of *aggregated values*, which are the sums of the expected cumulative rewards obtained for using each joint action, and finally an array of *Elo ratings*, one for each team.

The search tree is grown by following and repeating five steps.

(1) *Tree policy*. We descend the search tree by selecting promising teams and joint actions [2],
(2) *Parallel simulations*. We select a joint action for each team and run independent parallel random simulations until the planning horizon is reached.
(3) *Pairwise matches*. We then compare the rewards accrued after each simulation and do pairwise matches between all the teams. A team *wins* against another team if it has generated a larger reward. The Elo ratings are updated via equation 1.
(4) *Tree expansion*. We expend the search tree with the team having generated the largest reward after its random simulation.
(5) *Reward backpropagation*. Finally, we back-propagate this reward to the nodes visited during the Tree policy.

One iteration of those five steps is called a *playout*, and a repetition of multiple playouts constitutes the *MELO algorithm*. By running a sufficient number of playouts, one can expect that a node which was often visited during the Tree policy will store an accurate estimation of the value of each team and promising joint actions. A deterministic joint policy can be then extracted from the search tree by selecting, at each node, the best average joint action.

## 5 EXPERIMENTAL EVALUATION

As the problem of team reformation planning under uncertainty has not been studied before, we need new benchmarks to evaluate our approach. We introduce the *Disaster domain*, which is an extension of the *Firefighting domain*, a benchmark used in the evaluation of Dec-POMDP planning algorithms [14]. In this new scenario, illustrated on figure 2, a wild fire started and now threatens to destroy the houses of a neighborhood. Two types of agents, *excavators* and *firefighters*, each having their specific abilities, need to form and re-form in teams in order to clear the construction debris and fight the fire. Preliminary results show that the MELO algorithm is able

**Figure 2: The *Disaster domain*. An excavator is clearing the way to the first house while the fire truck is extinguishing the fire at the third house. The second house is blocked and needs to have its way cleared before the fire truck can go put out the fire.**

to quickly identify irrelevant teams and rule them out of the final joint policy. Teams composed of excavators only will rarely be used because it is more rewarding to extinguish a fire rather than to clear the access to a house. Further experiments need to be done in order to fully confirm our methods, as some of our tests showed that the Elo rating system can unfortunately also exclude good teams.

## 6 CONCLUSION

The Team-POMDP framework allows to model the process of dynamic teams adjusting to the situation as time goes on. The problem of team reformation planning under uncertainty represents a challenging but promising field of research for the years to come. Our work is an attempt dedicated at breaking the inherent complexity of the model, though more work remains to be done to efficiently compute optimal separable joint policies.

## REFERENCES


[1] Pritee Agrawal and Pradeep Varakantham. 2017. Proactive and Reactive Coordination of Non-dedicated Agent Teams Operating in Uncertain Environments. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*. AAAI Press, Palo Alto, CA, USA, 28–34.

[2] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47, 2-3 (2002), 235–256.

[3] Daniel S Bernstein, Shlomo Zilberstein, and Neil Immerman. 2000. The Complexity of Decentralized Control of Markov Decision Processes. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 32–37.

[4] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. 2012. A survey of Monte Carlo Tree Search methods. *IEEE Transactions on Computational Intelligence and AI in games* 4, 1 (2012), 1–43.

[5] Muthukumaran Chandrasekaran, Adam Eck, Prashant Doshi, and Leenkiat Soh. 2016. Individual Planning in Open and Typed Agent Systems. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence (UAI'16)*. AUAI Press, Arlington, VA, United States, 82–91.

[6] Jonathan Cohen, Jilles Steeve Dibangoye, and Abdel-Illah Mouaddib. 2017. Open Decentralized POMDPs. In *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, New York, NY, USA, 977–984.

[7] Jonathan Cohen and Abdel-Illah Mouaddib. 2018. Monte-Carlo Planning for Team Re-Formation Under Uncertainty: Model and Properties. In *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, New York, NY, USA, 458–465.

[8] Rémi Coulom. 2007. Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search. In *Computers and Games*, H. Jaap van den Herik, Paolo Ciancarini, and H. H. L. M. (Jeroen) Donkers (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 72–83.

[9] Jilles Steeve Dibangoye, Christopher Amato, Olivier Buffet, and François Charpillet. 2016. Optimally solving Dec-POMDPs as continuous-state MDPs. *Journal of Artificial Intelligence Research* 55 (2016), 443–497.

[10] Arpad E. Elo. 1978. *The rating of chessplayers, past and present*. Arco Pub., New York, NY, USA.

[11] Matthew E. Gaston and Marie DesJardins. 2005. Agent-organized Networks for Dynamic Team Formation. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '05)*. ACM, New York, NY, USA, 230–237.

[12] Ranjit Nair, Milind Tambe, and Stacy Marsella. 2003. Role Allocation and Reallocation in Multiagent Teams: Towards a Practical Analysis. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '03)*. ACM, New York, NY, USA, 552–559.

[13] Frans A. Oliehoek and Christopher Amato. 2016. *A Concise Introduction to Decentralized POMDPs* (1st ed.). Springer Publishing Company, Incorporated, New York, NY, USA.

[14] Frans A. Oliehoek, Matthijs T. J. Spaan, and Nikos Vlassis. 2008. Optimal and Approximate Q-value Functions for Decentralized POMDPs. *Journal of Artificial Intelligence Research* 32, 1 (2008), 289–353.

[15] Peter Stone, Gal A. Kaminka, Sarit Kraus, and Jeffrey S. Rosenschein. 2010. Ad Hoc Autonomous Agent Teams: Collaboration Without Pre-coordination. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI'10)*. AAAI Press, Palo Alto, CA, USA, 1504–1509.

[16] Milind Tambe. 1997. Towards flexible teamwork. *Journal of Artificial Intelligence Research* 7 (1997), 83–124.

[17] Paulo Trigo and Helder Coelho. 2005. The Multi-team Formation Precursor of Teamwork. In *Progress in Artificial Intelligence*, Carlos Bento, Amílcar Cardoso, and Gaël Dias (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 560–571.

[18] Rogier M. van Eijk, Frank S. de Boer, Wiebe van der Hoek, and John-Jules C. Meyer. 2000. Open Multi-Agent Systems: Agent Communication and Integration. In *Intelligent Agents VI. Agent Theories, Architectures, and Languages*, Nicholas R. Jennings and Yves Lespérance (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 218–232.