# Decentralized Multi-Agent Reinforcement Learning in Average-Reward Dynamic DCOPs[*]

# (Extended Abstract)

Duc Thien Nguyen[†], William Yeoh[‡], Hoong Chuin Lau[†],
Shlomo Zilberstein[⋆], Chongjie Zhang[*]
[†]School of Information Systems, Singapore Management University, Singapore 178902
[‡]Department of Computer Science, New Mexico State University, Las Cruces, NM 88003, USA
[⋆]School of Computer Science, University of Massachusetts, Amherst, MA 01003, USA
[*]CSAIL, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

## ABSTRACT

Researchers have introduced the *Dynamic Distributed Constraint Optimization Problem* (Dynamic DCOP) formulation to model dynamically changing multi-agent coordination problems, where a dynamic DCOP is a sequence of (static canonical) DCOPs, each partially different from the DCOP preceding it. Existing work typically assumes that the problem in each time step is decoupled from the problems in other time steps, which might not hold in some applications. In this paper, we introduce a new model, called *Markovian Dynamic DCOPs* (MD-DCOPs), where a DCOP is a function of the value assignments in the preceding DCOP. We also introduce a distributed reinforcement learning algorithm that balances exploration and exploitation to solve MD-DCOPs in an online manner.

## Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed AI

## Keywords

DCOP; Dynamic DCOP; MDP; Reinforcement Learning

## 1. INTRODUCTION AND BACKGROUND

*Distributed Constraint Optimization Problems* (DCOPs) are problems where agents need to coordinate their value assignments to maximize the sum of the resulting constraint utilities [2, 3, 5]. Unfortunately, DCOPs only model static problems or, in other words, problems that do not change over time. In many multi-agent coordination problems, various events that change the problem can occur. As a result, researchers have extended DCOPs to *Dynamic DCOPs*, where the problem can change over time. Researchers have thus far taken an *online* approach by modeling it as a sequence of (static canonical) DCOPs, each partially different from the DCOP preceding it, and solving it by searching for a new solution each time the problem changes. Existing work typically assumes that the problem in each time step is decoupled from the problems in other time steps, which might not hold in some applications.

Therefore, in this paper, we introduce a new model, called *Markovian Dynamic DCOPs* (MD-DCOPs), where the DCOP in the next time step is a function of the value assignments in the current time step. Similar to existing work on dynamic DCOPs, we assume that the agents in MD-DCOPs are not aware of the underlying transition functions and, thus, need to solve the problem in an online manner. Specifically, we introduce a reinforcement learning algorithm, the distributed RVI Q-learning algorithm, that uses a multi-arm bandit strategy to balance exploration (learning the underlying transition functions) and exploitation (taking the currently believed optimal joint action). We empirically evaluate them against an existing multi-arm bandit DCOP algorithm on dynamic DCOPs.

A DCOP is defined by $\langle \mathcal{X}, \mathcal{D}, \mathcal{F}, \mathcal{A}, \alpha \rangle$, where $\mathcal{X} = \{x_1, \ldots, x_n\}$ is a set of *variables*; $\mathcal{D} = \{D_1, \ldots, D_n\}$ is a set of finite *domains*, where $D_i$ is the domain of variable $x_i$; $\mathcal{F} = \{f_1, \ldots, f_m\}$ is a set of *utility functions* (also called *constraints*), where each $k$-ary utility function $f_i : D_{i_1} \times D_{i_2} \times \ldots \times D_{i_k} \mapsto \mathbb{N} \cup \{-\infty, 0\}$ specifies the utility of each combination of values of variables in its *scope* (i.e., $x_{i_1}, \ldots, x_{i_k}$); $\mathcal{A} = \{a_1, \ldots, a_p\}$ is a set of *agents*; and $\alpha : \mathcal{X} \rightarrow \mathcal{A}$ maps each variable to one agent. A *solution* is a value assignment for all variables. Its utility is the evaluation of all utility functions on that solution. The goal is to find a utility-maximal solution.

## 2. MARKOVIAN DYNAMIC DCOPs

At a high level, a *Markovian Dynamic DCOP* (MD-DCOP) can be visualized as a sequence of (static canonical) DCOPs with one (static canonical) DCOP associated with each time step. The variables $\mathcal{X}$, domains $\mathcal{D}$, agents $\mathcal{A}$, and ownership mapping $\alpha$ of the initial DCOP remains unchanged across all time steps, but the reward functions $\mathcal{F}$ *can* change and is a function of the global joint state $\mathbf{s}$ at the current time step. The problem transitions from one global joint state to another in the next time step according

[*]This research is partially supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office and by the National Science Foundation under grant number IIS-1116917.

**Appears in:** *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

to a pre-defined joint transition probability function, which is a function of the values of the agents in the current time step. In this paper, we assume that the agents do not know this underlying joint transition probability function.

In more detail, an MD-DCOP is defined by a tuple $\langle \mathbf{S}, \mathbf{D}, \mathbf{P}, \mathbf{F} \rangle$, where

**S** is the finite set of global joint states. $\mathbf{S} = \times_{1 \leq i \leq m} \mathbf{S}_i$, where $\mathbf{S}_i$ is the set of local states of reward function $f_i \in \mathcal{F}$. Each global joint state $\mathbf{s} \in \mathbf{S}$ is defined by $\langle \mathbf{s}_1, \ldots, \mathbf{s}_m \rangle$, where $\mathbf{s}_i \in \mathbf{S}_i$.

**D** is the finite set of global joint values. $\mathbf{D} = \times_{1 \leq i \leq n} D_i$, where $D_i \in \mathcal{D}$ is the set of local values of variable $x_i$. Each global joint value $\mathbf{d} \in \mathbf{D}$ is defined by $\langle d_1, \ldots, d_n \rangle$, where $d_i \in D_i$. We also use the notation $\mathbf{D}_i$ to denote the set of local joint values of variables $x_{i_1}$ through $x_{i_k}$ that are in the scope of reward function $f_i \in \mathcal{F}$. Each local joint value $\mathbf{d}_i \in \mathbf{D}_i$ is defined by $\langle d_{i_1}, d_{i_2}, \ldots, d_{i_k} \rangle$, where $d_{i_j} \in D_{i_j}$.

**P** is the finite set of joint transition probability functions that assume conditional transition independence. $\mathbf{P} = \times_{\mathbf{s}, \mathbf{s}' \in \mathbf{S}, \mathbf{d} \in \mathbf{D}} P(\mathbf{s}' \mid \mathbf{s}, \mathbf{d})$, where $P(\mathbf{s}' \mid \mathbf{s}, \mathbf{d}) = \Pi_{1 \leq i \leq m} P_i(\mathbf{s}'_i \mid \mathbf{s}_i, \mathbf{d}_i)$ is the probability of transitioning to joint state $\mathbf{s}'$ after taking joint value $\mathbf{d}$ in joint state $\mathbf{s}$. In this paper, we assume that the underlying joint transition probably functions are *not known* to the agents a priori.

**F** is the finite set of joint reward functions. $\mathbf{F}(\mathbf{s}, \mathbf{d}) = \sum_{1 \leq i \leq m} f_i(\mathbf{s}_i, \mathbf{d}_i)$, where $f_i(\mathbf{s}_i, \mathbf{d}_i)$ is the reward of taking joint value $\mathbf{d}_i$ in joint state $\mathbf{s}_i$.

A solution to an MD-DCOP is a global joint policy $\Pi : \mathbf{S} \mapsto \mathbf{D}$ that maps each global joint state $\mathbf{s} \in \mathbf{S}$ to a global joint value $\mathbf{d} \in \mathbf{D}$. The objective of an MD-DCOP is for the agents to assign a sequence of values to their variables (to learn the underlying transition probability function and explore the state space) and converge on a global joint policy that together maximizes the expected average reward:

$$\lim_{T \to \infty} \frac{1}{T} \mathbb{E} \left[ \sum_{t=0}^{T-1} \sum_{i=1}^{m} f_i(\mathbf{s}_i, \mathbf{d}_i^t) \right] \tag{1}$$

where $\mathbf{d}_i^t$ is the local joint value for local state $\mathbf{s}_i$ at time step $t$.

## 3. DISTRIBUTED RVI Q-LEARNING

Since the underlying transition probability functions of MD-DCOPs are not known to the agents a priori, there is a clear need for algorithms that perform exploration vs. exploitation tradeoffs. In this paper, we explore reinforcement learning methods to solve this problem. Specifically, we extend the (centralized) RVI Q-learning algorithm [1] to solve MD-DCOPs in a distributed way:

**Step 1**: For each reward function $f_i \in \mathcal{F}$, the lower priority agent in the scope of the reward function will initialize (say to 0) the current time step $t$, local Q values $Q_i^t(\hat{\mathbf{s}}_i, \hat{\mathbf{d}}_i)$ for all local states $\hat{\mathbf{s}}_i \in \mathbf{S}_i$ and local joint values $\hat{\mathbf{d}}_i \in \mathbf{D}_i$.

**Step 2**: Let the current local state be $\mathbf{s}_i$ and the global joint state be $\mathbf{s}$. Choose the local joint value $\mathbf{d}_i$ based on the multi-arm bandit exploration strategy described above. Then, the immediate joint reward of choosing this value in this state is $f_i(\mathbf{s}_i, \mathbf{d}_i)$, and let the next resulting local state be $\mathbf{s}'_i$.

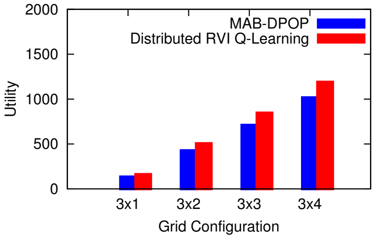

**Figure 1: Experimental Results**

**Step 3**: Broadcast the new local state $\mathbf{s}'_i$, and let the new global joint state be $\mathbf{s}'$.

**Step 4**: Find the DCOP solution that maximizes $\sum_{i=1}^{m} Q_i^t(\mathbf{s}', \mathbf{d}'_i \in \mathbf{d}')$, and then update the local Q values according to

$$\begin{aligned} Q_i^{t+1}(\mathbf{s}_i, \mathbf{d}_i) = Q_i^t(\mathbf{s}_i, \mathbf{d}_i) + \gamma(t) \big[ & f_i(\mathbf{s}_i, \mathbf{d}_i) \\ & + Q_i^t(\mathbf{s}'_i, \mathbf{d}'_i \mid \mathbf{d}'_i \in \operatorname*{argmax}_{\mathbf{d}' \in \mathbf{D}} Q^t(\mathbf{s}', \mathbf{d}')) \\ & - Q_i^t(\mathbf{s}_i, \mathbf{d}_i) - Q_i^t(\mathbf{s}_i^0, \mathbf{d}_i^0) \big] \end{aligned} \tag{2}$$

**Step 5**: Repeat Steps 2 through 4 until convergence.

## 4. RESULTS AND CONCLUSIONS

We compared the distributed Q-learning algorithm to Multi-Arm Bandit DCOP (MAB-DCOP) [4], a regret-minimizing algorithm that seeks to maximize the expected cumulative reward over the time horizon in a DCOP with reward uncertainty. We ran our experiments on a 64 core Linux machine with 2GB of memory and evaluated the algorithms on a sensor network domain. Figure 1 shows some preliminary results, where our Distributed Q-learning algorithm is shown to find better solutions than MAB-DCOP, which is not surprising as MAB-DCOP was not designed to solve MD-DCOPs and thus does not exploit the assumption that the underlying transitions are Markovian.

In this paper, we take a first step towards capturing the inter-dependence between problems in subsequent time steps in a dynamic DCOP via the MD-DCOP model. We also introduced a distributed reinforcement learning algorithm to solve this problem, and showed that it outperforms MAB-DPOP in a range of sensor network problems.

## 5. REFERENCES

[1] J. Abounadi, D. Bertsekas, and V. Borkar. Learning algorithms for markov decision processes with average cost. *SIAM Journal on Control and Optimization*, 40(3):681–698, 2001.

[2] P. Modi, W.-M. Shen, M. Tambe, and M. Yokoo. ADOPT: Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161(1–2):149–180, 2005.

[3] A. Petcu and B. Faltings. A scalable method for multiagent constraint optimization. In *Proceedings of IJCAI*, pages 1413–1420, 2005.

[4] R. Stranders, F. Delle Fave, A. Rogers, and N. Jennings. DCOPs and bandits: Exploration and exploitation in decentralised coordination. In *Proceedings of AAMAS*, pages 289–297, 2012.

[5] W. Yeoh and M. Yokoo. Distributed problem solving. *AI Magazine*, 33(3):53–65, 2012.