Modeling multi-objectivization mechanism in multi-agent domain

1 Kousuke Nishi
Department of Urban Environment Systems
Chiba University
Chiba, Japan
afaa1996@chiba-u.jp

2nd Sachiyo Arai
Department of Urban Environment Systems
Chiba University
Chiba, Japan
arai@tu.chiba-u.ac.jp

***Abstract*—Many real-world tasks require making sequential decisions that involve multiple conflicting objectives. Furthermore, there exist multiple decision-makers, called multiagent, each of whom pursues its own profit. Thus, each agent should take into account the effect of other agents' decisions to reach a point of compromise. For example, each agent decides with thought of other agents' behavior in the decision of selecting the faster driving route to the destination, selecting a supermarket checkout line, and so on. For solving a sequential multi-objective decision problem, a multi-objective reinforcement learning (MORL) approach has been investigated.**

However, current research on MORL cannot deal with the multi-agent system where existing agents are influenced one another. Therefore, in this study, we expand the conventional multi-objective reinforcement learning by introducing the idea of multi-objectivization with dynamic weight setting of other decision-makers. In an experiment, our proposed model with dynamic weight can express the cooperative behaviors that seems to be considered other decision-makers in the multiagent environment.

***Index Terms*—multi-objective decision-making, reinforcement learning, dynamic weight, multi-objectivization**

I. Introduction

Many of the decision-making is multi-objective decision-making. For example, when driving to a destination, a decision-maker determines the route, considering speed and cost. However, since there are multiple decision-makers in the real world, they decide the route also considering the crowded situation. The objectives considered in multi-objective decision-making can be roughly divided into "individual objectives" such as speed or cost and "objectives considering others" such as crowded situation. In the real world, as decision-makers have the objective considering others not only individual objectives, each decision-maker often takes cooperative actions while interacting each other.

Multi-objective decision-making can be generally modeled by Multi-Objective Reinforcement Learning(MORL). However, because conventional MORL assumes only one decision-maker, it is difficult to model decision-making considering others. Therefore, in this study, we expand conventional MORL, model the phenomena of the whole system where decision-makers interact with each other. In expanding MORL, we use two approaches, multi-objectivization, and dynamic weight setting. The first multi-objectivization is the process of replacing a single objective problem with a multi-objective problem to solve the original problem better. In multi-objectivization, there are two ways, decomposing the original objective or adding the extra objectives. This study models phenomena in the real world by adding the objectives considering others to individual objectives by multi-objectivization. The second dynamic weight is the altering weight of a decision-maker depending on the situation. In the real world, the influence of each other causes the weight to alter sequentially. Therefore, introducing dynamic weight setting enables the modeling of phenomena. In the experiment, we used the Deep Sea Treasure environment, which is a benchmark of MORL. Here, by introducing multi-objectivization and dynamic weights, we could confirm the cooperative behavior decision-makers disperse.

The contribution of this paper is that proposed method can model selfish or cooperative action, and balanced action of each agent. Then that enables modeling of complex behavior in multi-agent domain.

II. Preliminaries

A. Reinforcement Learning

Reinforcement Learning(RL) is method for obtaining optimal control performance through trial and error in an unknown environment[1]. RL is an algorithm that assumes MDP.

1) MDP: The Markov Decision Process (MDP) is a mathematical model for the dynamic optimization of stochastic systems with Markov properties in state transitions. MDP is defined as $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma \rangle$. $\mathcal{S}$ represents a state set, $\mathcal{A}$ represents an action set, $\mathcal{R}$ represents a reward function, $\mathcal{P}$ represents a transition probability set, and γ represents a discount rate. At time t, the agent observes the state $s_t \in S$, and selects the action $a_t \in A$ based on its own policy π_t. After that, at time $(t+1)$, s_t and a_t transition to the next state s_{t+1} by a certain probability, and reward r_{t+1} is obtained. Generate a value function from the received reward, and use that value to learn a policy π, which is the probability of selecting a possible action a from the state s.

2) Optimal Policy: The agent does not have know reward function $\mathcal{R}$ and state transition probability $\mathcal{P}$ beforehand. About the state when the time is t, the evaluation is made based on how much the reward can be obtained by the end

of the task. The state value function represents the expected reward when following the policy π from the state s.

$$V^{\pi}(s) = E_{\pi}\left\{ \sum_{k=0}^{\infty} \gamma^{k} r_{t+k+1} \;\middle|\; s_t = s \right\} \tag{1}$$

The value function of state action pair(s, a) is defined $Q(s, a)$. $Q^{\pi}(s, a)$ in Eq(2) represents the expected reward when following the policy π after taking action in state s. The optimal action value function $Q^{*}(s, a)$ is the expected value of discounted reward for an infinite period when continuing to take the optimal policy after executing action a in state s.

$$Q^{\pi}(s, a) = E_{\pi}\left\{ \sum_{k=0}^{\infty} \gamma^{k} r_{t+k+1} \;\middle|\; s_t = s, a_t = a \right\} \tag{2}$$

$$Q^{*}(s, a) = \sum_{s'} P^{a}_{ss'}[R^{a}_{ss'} + \gamma \max_{a'} Q^{*}(s', a')] \tag{3}$$

B. Multi-objective Reinforcement Learning

Multi-objective reinforcement learning is an algorithm for finding Pareto optimal policies assuming MOMDP.

1) MOMDP: Multi-objective Markov decision process(MOMDP) is an environment that extends MDP to multi-objective problems, defined as $\langle \mathcal{S}, \mathcal{A}, \boldsymbol{R}, \mathcal{P}, \gamma \rangle$. The difference of MDP is that $\boldsymbol{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{n}$ is reward vector, n is the number of objectives.

2) Pareto Optimal Policy: In MOMDP, since the reward is a vector, the state value function is also a vector. If the following (4) formula holds, π is a Pareto optimal policy[2].

$$\forall i, V_i^{\pi} \geq V_i^{\pi'} \wedge \exists i, V_i^{\pi} > V_i^{\pi'} \tag{4}$$

C. Multi-objectivization

The multi-objectivization of a problem is the conversion of a single-objective problem to a multi-objective problem to improve performance on the original objective, as measured by solution quality, time to solution, or some other measure[3]. There are two approaches in multi-objectivization : either through the decomposition of the original single objective or through the addition of extra objectives. We define MOMDP p, π belonging to the Pareto front $\mathscr{S}_p^{*}$, and expected discount cumulative reward J^{π}, the following relationship holds.

$$\pi \in \mathscr{S}_p^{*} \Leftrightarrow \exists_o \in O_p, \forall \pi' \in \Pi_p : J_o^{\pi} + \epsilon_o \geq J_o^{\pi'} \tag{5}$$

ϵ_o is a constant representing the preference of the system designer, O_p is the objectives set, and Π_p is the policies set that can be obtained by MOMDP. A comparison of MOMDP and CMOMDP is shown in Fig.1. The blue point in Fig.1 is Pareto optimal policy. The difference with MOMDP is that there is almost no trade-off between objectives. Formally,

$$p \in \text{CMOMDP} \Leftrightarrow \forall_o \in O_p : \max_{\pi \in \mathscr{S}_p^{*}} (J_o^{\pi}) - \min_{\pi' \in \mathscr{S}_p^{*}} (J_o^{\pi'}) \leq \epsilon_o \tag{6}$$

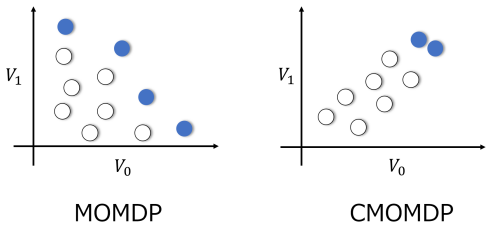

Fig. 1. MOMDP and CMOMDP

III. Related Work

This chapter introduces related work on multi-objectivization and dynamic weight. Multi-objectivization is defined by Knowles, and Knowles applied this approach to the local search method, the hill climbing method [4]. By introducing multi-objectivization to the hill climbing method, more freedom to explore, and it is possible to obtain a more optimal solution without falling into a local solution. Multi-objectivization has been mainly used to improve computational efficiency in the study of evolutionary computation. In reinforcement learning, Brys et al. introduced this approach to speed up learning[5]. Brys et al. applied multi-objectivization in two reinforcement learning tasks, the path search problem, and the Mario game. That paper, by combining the correlated objectives well, speeds up learning, improve the performance, and demonstrated the usefulness of multi-objectivization [7]. In dynamic weight, Kallstrom et al. adjusted the agent's weight to construct an agent-based simulation that meets the needs of the user[8]. Also, Brys et al. proposed an adaptive objective selection that considers the most reliable objective in each state if action is a probability distribution[9]. So far in dynamic weight, there is no study on the assumption that the weight alters under the influence of other agents, as far as we know. Therefore, in this paper, we propose dynamic weight by the actions of other agents.

IV. Target Problem

The types of decision-making can be classified into three axes: the number of objectives, decision-making, decision-makers. This paper deals with multi-objective sequential decision-making problems in a multi-agent domain. In general, multi-objective sequential decision-making problems can be solved by multi-objective reinforcement learning, assuming MOMDP. Table Ⅱ shows typical methods using multi-objective reinforcement learning. In this study, we use the Weighted Sum method, it is a single-policy approach that obtains one policy in learning, and the scalarization function linear[10].

TABLE I
ALGORITHM OF MULTI-OBJECTIVE REINFORCEMENT LEARNING

	linear	non-linear
single-policy	Weighted Sum	Chebyshev scalarization
multiple-policy	Convex Hull Value Iteration	Pareto Front Value Iteration

V. Approach

A. Dynamic Weight

We define objectives set O owned by a decision-maker; objectives set O_{old} to consider in decision-making. Also, define objectives set O_{add} added by multi-objectivization, and objectives set O_{new} to newly consider in decision-making. The relation of each objective set is shown in the equation (7).

$$O_{old} \subset O + O_{add} \subset O = O_{new} \subset O \tag{7}$$

The relationship between w_{old}, w_{add}, and w_{new} by multi-objectivization is shown below. The weight w_{old} for the objectives originally considered is indicated by multiplying the tenacity α_j. When a decision-maker consider a new objective, the value of α_j is large if you cling to objective j. Otherwise, the value is smaller.w_{add} is a function of state. In this paper, this state relates to other agents.

$$w_{new} = \begin{cases} w_{j,old} \leftarrow \alpha_j w_{j,old} & (\sum_j \alpha_j = 1) \\ w_{add} = f(s) & (\sum w_{new} = w_{old} + w_{add} = 1) \end{cases}$$

B. Multi-objectivization

We introduce the proposed algorithm based on MORL about multi-objectivization by other agents. The proposed algorithm is shown in Algorithm 1. First of all, there are three inputs in this algorithm, the first is MDP or MOMDP, the second is agents set, and the third is a weight set with the weight of each agent as an element. In the forth and subsequent lines are the learning of each agent. In the rough learning flow, the turn in which each agent learns is determined in line 5, and learning is performed in each turn. Therefore, in this approach, two or more agents do not learn at the same time. The subscript i of $agent_i^\tau$ is the identification number of the agent, and the superscript τ is the turn in which $agent_i$ learns. One episode is that all agents learn once. When one episode is over, the turn in which each agent learns is determined again, and the learning is performed again in that order. This is repeated until the E episode. The 12th and 13th line of this algorithm is the most critical point during learning. Here, by transforming into a CMOMDP through multi-objectivization by $\pi^{agent_i^\tau}$ the dynamic weight described in Section V is performed.

Algorithm 1 Proposed algorithm

1: **Input** : MDP or MOMDP $Agent = \{agent_1, agent_2 ... agent_n\}$
$\boldsymbol{W} = \{\boldsymbol{w}_{old}^{agent_1}, \boldsymbol{w}_{old}^{agent_2}, ..., \boldsymbol{w}_{old}^{agent_n}\}$
2: E : The maximum number of episodes
3: M : The number of objectives N : The number of agents
4: **for** $h = 0$ to E **do**
5: Randomly determine the turn τ of $agent_i$
6: **for** $\tau = 1$ to N **do**
7: **repeat**
8: $r(\boldsymbol{s}, a) = \sum_{j=1}^{M} w_{j,old}^{agent_i^\tau} r_j(\boldsymbol{s}, a)$
9: $Q^* = Q^{agent_i^\tau}$
10: $Q^*(\boldsymbol{s}, a) \leftarrow \sum_{\boldsymbol{s}'} p_{\boldsymbol{ss}'}^a [r_{\boldsymbol{ss}'}^a + \gamma \max_{a'} Q^*(\boldsymbol{s}', a')]$
11: **until** S is terminal
12: **Output** : $\pi^{agent_i^\tau}$
13: Transforming into a CMOMDP through multi-objectivization
14: Altering weight $\boldsymbol{w}_{old}^{agent^{\tau+1}} \leftarrow \boldsymbol{w}_{new}^{agent^{\tau+1}}$
15: **end for**
16: **end for**

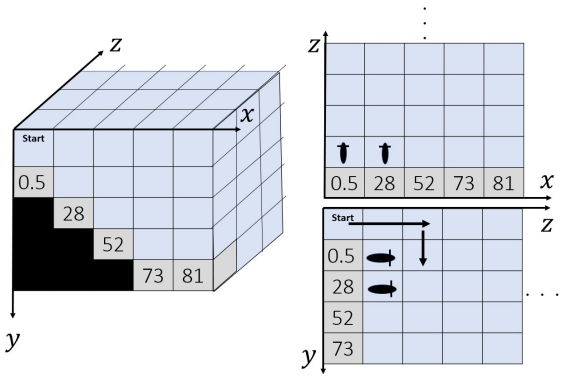

Fig. 2. Experiment environment

VI. Experiment

A. Experimental Setting

We use a Deep Sea Treasure(DST) environment[11] expanded in three dimensions for experiments. The experimental environment is shown in Fig.2. The right of the Fig.2 is a figure looking from the top and the right. The blue cells represent the sea, the black cells represent the seabed, and gray cells represent treasures. The action of the submarine is to move one cell in six direction. In this experiment, the number of agents operating the submarines is set to 30, and when one submarine gets a treasure, the next submarine starts searching from the start. If a submarine gets a treasure that has already been gotten by more than one, an extra step is added in the z-direction. Table II shows the correspondence between reward and weight of each objective. In conventional DST environment, the objective of the agent is to minimize the number of steps to the treasure and to maximize the value of the treasure. In this experiment, the minimization of the number of steps is decomposed into the minimization of the number of steps (x,y-axis) and the number of steps(z-axis). The discount rate of γ is 0.95.

B. Preliminary Experiment

In the preliminary experiment, we studied which treasures to select by altering the weights of w_1 and w_2 exhaustively. The treasures from the start are defined $X_1, X_2, ..., X_5$, and the results are shown in the table III. In this experiment, under this result, $\boldsymbol{w}_{old}^{agent_i}$ of each agent is set with the probability of $P(X)$ shown below the table.

TABLE III
Range of weights for each treasure

	$X_1(0.5)$	$X_2(28)$	$X_3(52)$	$X_4(73)$	$X_5(81)$
$w_{1,old}$	[0.93,1]	[0.91,0.92]	[0.89,0.9]	[0.81,0.88]	[0,0.8]
$w_{2,old}$	[0,0.07]	[0.08,0.09]	[0.1,0.11]	[0.12,0.19]	[0.2,1]
$P(X)$	0.4	0.3	0.1	0.1	0.1

C. Evaluation Method

$\boldsymbol{w}_{add}^{agent_i}$ is a function of state. This experiment divides the state into the following three cases. The left of Fig.3 shows the relationship between the turn the agent learns and the weight w_{add}. The first is the Crowded or uncrowded of each treasure. Crowded or uncrowded of the treasure determine by the number of agents that got the treasure. If one treasure

TABLE II
REWARD AND WEIGHT OF EACH OBJECTIVE

O_{old}	O_{new}	reward	weight
minimize the number of steps	minimize the number of steps(x,y-axis)	-1 each step	$w_{1,old}$
	minimize the number of steps(z-axis)(O_{add})		w_{add}
maximize the value of treasures	maximize the value of treasures	value of treasure or 0	$w_{2,old}$

Fig. 3. Experiment result

is crowded, set the weight w_{add} to a constant value. This case, after one treasure is crowded, the value of the weight is constant. The second sets the state as own turn τ. η is proportional constant, and N is the number of agents. This weight set is a setting in which the weight w_{add} increases in proportion to its own turn. The third is the variance of other agents. The agent observes the variance of other agents who have selected each treasure. If the variance exceeds a certain value, set the weight w_{add} to a constant value.

1) Crowded or uncrowded

$$w_{add}^{agent_i} = \begin{cases} k_1 & (crowded) \\ 0 & (uncrowded) \end{cases}$$

2) Own turn

$$w_{add}^{agent_i} = \eta \frac{\tau}{N}$$

3) Variance of other agents

$$w_{add}^{agent_i} = \begin{cases} k_2 & (\sigma_{\tau-1}^2 > \iota) \\ 0 & (else) \end{cases}$$

In this experiment, we set 5 agents as the border whether crowded or uncrowded in the first setting. η in second setting is 0.5, ι in third setting is 1. In this experiment, we evaluate the agent's variance value for the two objectives before multi-objectivization and the three objectives proposed in this study. Here, if the variance value is small, it can be said that each agent considers other agents.

D. Results

The experimental results show at the right of Fig.3. The horizontal axis of the graph represents the turn of learning, and the vertical axis represents the variance in that turn. We divided the setting of the state and the weight into three types and conducted experiments. From these experimental results, the agents dispersed and make them uncongested in each case. The acquired behavior of agent with our proposed method, considerable and cooperative behaviors have been emerged. Specifically, when the nearer location of treasure was already occupied and crowded, agent seems to decide to take farther treasure to avoid crowds.

VII. CONCLUSION

This study deals with sequential and multi-objective decision environment where exists multiple decision-makers interact with each other. In this study, we expand MORL by two approaches, multi-objectivization and dynamic weight setting. We added objectives considering others thorough multi-objectivization and proposed a dynamic weight setting depending on others. In the experiment, we introduced multi-objectivization by adding other agents in DST environment, which is conventionally single agent environment. Moreover, by adding the dynamic weight setting according to the selection of other agents, we realized the cooperative action which considered other agents.

REFERENCES

[1] Sutton, Richard S., and Andrew G. Barto. Introduction to reinforcement learning. Vol. 135. Cambridge: MIT press, 1998.
[2] Roijers, Diederik M., et al. "A survey of multi-objective sequential decision-making." Journal of Artificial Intelligence Research 48 (2013): 67-113.
[3] Brys, Tim, et al. "Multi-objectivization and ensembles of shapings in reinforcement learning." Neurocomputing 263 (2017): 48-59.
[4] Knowles, et al. "Reducing local optima in single-objective problems by multi-objectivization." International Conference on Evolutionary Multi-Criterion Optimization. Springer, Berlin, Heidelberg, 2001.
[5] Brys, Tim, et al. Multi-objectivization in reinforcement learning. Technical Report AI-TR-13-354, AI Lab, Vrije Universiteit Brussel, 2013.
[6] Brys, Tim, et al. "Combining multiple correlated reward and shaping signals by measuring confidence." Twenty-Eighth AAAI Conference on Artificial Intelligence. 2014.
[7] Brys, Tim, et al. "Multi-objectivization of reinforcement learning problems by reward shaping." 2014 international joint conference on neural networks (IJCNN). IEEE, 2014.
[8] K?llstr?m, Johan, and Fredrik Heintz. "Tunable Dynamics in Agent-Based Simulation using Multi-Objective Reinforcement Learning."
[9] Brys, Tim, et al. "Adaptive objective selection for correlated objectives in multi-objective reinforcement learning." Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems. International Foundation for Autonomous Agents and Multiagent Systems, 2014.
[10] Van Moffaert, et al. "Scalarized multi-objective reinforcement learning: Novel design techniques." 2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL). IEEE, 2013.
[11] Peter Vamplew, Richard Dazeley, Adam Berry, Rustam Issabekov, and Evan Dekker: "Empirical evaluation methods for multiobjective reinforcement learning algorithms", Machine learning, 84(1-2), pp. 51-80, 2011