# On the Rationality of Profit Sharing in Multi-agent Reinforcement Learning

Kazuteru Miyazaki
National Institution for Academic Degrees
teru@niad.ac.jp

Shigenobu Kobayashi
Tokyo Institute of Technology
kobayasi@dis.titech.ac.jp

**Abstract**

*Reinforcement learning is a kind of machine learning. It aims to adapt an agent to an unknown environment according to rewards. Traditionally, from theoretical point of view, many reinforcement learning systems assume that the environment has Markovian properties. However it is important to treat non-Markovian environments in multi-agent reinforcement learning systems. In this paper, we use Profit Sharing (PS) as a reinforcement learning system and discuss the rationality of PS in multi-agent environments. Especially, we classify non-Markovian environments and discuss how to share a reward among reinforcement learning agents. Through cranes control problem, we confirm the effectiveness of PS in multi-agent environments.*

## 1. Introduction

To achieve cooperation in multi-agent systems is a very desirable goal. Recently, an approach to realize cooperation by reinforcement learning is notable. There is much literature on multi-agent reinforcement learning. Previous works [1, 2] compare Profit Sharing (PS) [3] with Q-learning [6] in the pursuit problem [1] and the cranes control problem [2]. These papers [1, 2] claim that PS is suitable for multi-agent reinforcement learning systems.

In multi-agent environments where there is no negative reward, it is important to share a reward among all agents. Conventional work has used ad hoc sharing schemes. Though reward sharing may contribute to improve system behavior, it is possible to get no reward. Therefore, it is important to preserve the *rationality condition* that expected reward per an action is larger than zero ($\frac{expected\ reward}{an\ action} > 0$).

In this paper, we aim to preserve the rationality condition in multi-agent environments where there is no negative reward. We focus on the *Rationality Theorem of PS* [4] and analyze how to share a reward among all PS agents. We show the necessary and sufficient condition to preserve the rationality condition in multi-agent PS systems.

Section 2 shows the rationality theorem of PS. Section 3 discusses the rationality of PS in multi-agent systems. Section 4 shows an numerical example. Section 5 is conclusion.

## 2. Rationality of Profit Sharing

### 2.1. Target Environments

Consider an agent in some unknown environment. The agent senses the environment and performs an action. The environment gives a (positive) *reward* to the agent as a result of some sequence of action. The agent senses a set of discrete attribute-value pairs and performs an action in some discrete varieties. A pair of a sensory input and an action is called a *rule*. A scalar weight, that

represents the importance, is attached to each rule. The weight of rule $xa$ is denoted as $\omega_{xa}$. The function that maps sensory inputs to actions is called a *policy*. We call a policy *rational* if and only if expected reward per an action is larger than zero.

We call a sequence of rules selected between rewards an *episode*. PS reinforces rules on an episode at once. We call a function that shares a reward among rules on an episode a *reinforcement function*. $f_i$ denotes a reinforcement value for the rule selected at $i$ step before a reward is acquired. We assume that the weight of each rule is reinforced by $\omega_{r_i} = \omega_{r_i} + f_i$ for an episode $(r_W \cdots r_i \cdots r_2 \cdot r_1)$ where $W$ denotes the length of an episode called *reinforcement interval*. We call a subsequence of an episode a *detour* when the sensory input of the first selecting rule and the sensory output of the last selecting rule are the same though both rules are different. The rules on a detour may not contribute to get any reward. We call a rule *ineffective* if and only if it always exists on detours of any episode. Otherwise, a rule is called *effective*. Ineffective rules should not be reinforced when they conflict with effective rules.

If an effective rule is always on a detour, it is called a *type 2 confusion*. Markov Decision Process (MDP), that are treated by many reinforcement learning systems, does not contain any type 2 confusion. There is no type 2 confusion in the class where all state transitions that have been experienced do not disappear. The class contains MDP. Furthermore, it is more widely class that all state transitions that have been experienced by an effective rule do not disappear. In this paper, we show the *Rationality Theorem of PS* that guarantees the acquisition of a rational policy in the class where there is no type 2 confusion.

### 2.2. Rationality Theorem of Profit Sharing

We know the *Rationality Theorem* of PS as follows [4];

*Theorem 1 (Rationality Theorem of PS)*
*PS can learn a rational policy in the class where there is no type 2 confusion if and only if*

$$\forall i = 1, 2, ..., W. \quad L \sum_{j=i}^{W} f_j < f_{i-1}, \tag{1}$$

*where $L$ is an upper bound of the number of conflicting effective rules, and $W$ is an upper bound of the length of episodes.* □

It is a necessary and sufficient condition to learn a rational policy in the class where there is no type 2 confusion. We cannot know the number of $L$ in general. However, in practice, we can set $L = M - 1$ where $M$ is the number of actions. There are many functions that satisfy theorem 1.

## 3. Profit Sharing in Multi-agent Reinforcement Learning

### 3.1. Multi-agent Reinforcement Learning

We can guarantee the rationality by theorem 1 in multi-agent reinforcement learning systems if there is no type 2 confusion. The number of rules that have type 2 confusion is not stable since several agents may learn at the same time. We introduce an *indirect reward* to reduce a type 2 confusion.

Though an indirect reward may contribute to improve system behavior, it is possible to get no reward. Therefore, it is important to preserve the rationality condition. We show the *Rationality Theorem of Indirect-reward* to guarantee the rationality in multi-agent reinforcement learning when there is an indirect reward.

### 3.2. Rationality of Indirect-reward

#### 3.2.1. Problem Formulation

Consider $n$ ($n > 0$) agents in an unknown environment. At each discrete time step, *agent* $i(i = 1, 2, ..., n)$ is selected from $n$ agents based on the selection probabilities $P_i(P_i > 0, \Sigma_{i=1}^{n} P_i = 1)$, and it senses the environment and performs an action. The agent senses a set of discrete attribute-value pairs and performs an action in $M$ discrete varieties.

When the $n'$th agent ($0 < n' \leq n$) has a *special sensory input* on condition that $(n' - 1)$ agents have special sensory inputs at some time step, the $n'$th agent gets a *direct reward* $R$ ($R > 0$) and the other $(n-1)$ agents get an *indirect reward* $\mu R$ ($\mu \geq 0$). We call the $n'$th agent the *direct-reward agent* and the other $(n-1)$ agents *indirect-reward agents*. We do not have any information about the $n'$ and the *special sensory input*. Furthermore, nobody (including *reward designers*) knows whether $(n-1)$ agents except for the $n'$th agent are important or not. A set of $n'$ agents that are necessary for getting a direct reward is called the *goal-agent set*. In order to preserve the *rationality condition* that expected reward per an action is larger than zero, all agents in a goal-agent set must learn a rational policy.

When a reward $f_0$ is given to the agent, we use the following reinforcement function that satisfies the *Rationality Theorem of PS* [4],

$$f_n = \frac{1}{M} f_{n-1}, \quad n = 1, 2, \cdots, W_a - 1. \tag{2}$$

where,($f_0$,$W_a$) is ($R$,$W$) for the direct-reward agent and ($\mu R$,$W_0$ ($W_0 \leq W$)) for indirect-reward agents.

#### 3.2.2. Rationality Theorem of Indirect-reward

An ineffective rules should not be reinforced when they conflict with an effective rule. When $\mu > 0$, an ineffective rule may be the most reinforced rule in some state. Therefore, it is important to suppress all ineffective rules in the most reinforced rule in all states. We know the necessary and sufficient condition about the range of $\mu$ to suppress all ineffective rules in some goal-agent set as follows [5];

*Theorem 2 (Rationality Theorem of Indirect-reward)*
*Any ineffective rule in some goal-agent set can be suppressed if and only if*

$$\mu < \frac{M - 1}{M^W (1 - (\frac{1}{M})^{W_0})(n-1)L}, \tag{3}$$

*where $M$ is the maximum number of conflicting rules in the same sensory input, $L$ is the maximum number of conflicting effective rules, $W$ is the maximum episode length of a direct-reward agent, $W_0$ is the reinforcement interval of indirect-reward agents, and $n$ is the number of agents.* □

We cannot know the number of $L$ in general. However, in practice, we can set $L = M - 1$. Furthermore, we cannot know the number of $W$ in general. However, in practice, we can set $\mu = 0$ if the length of an episode is larger than $W$. If we set $L = M - 1$ and $W_0 = W$, theorem 2 is simplified as follows;

$$\mu < \frac{1}{(M^W - 1)(n-1)}. \tag{4}$$

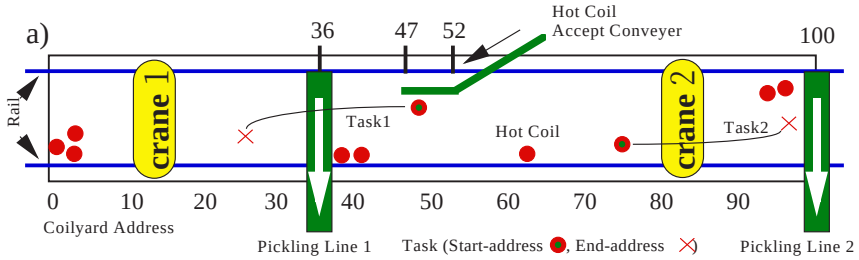
### 3.3. Toward Type 2 Confusion Reduction Theorem

Theorem 2 is a necessary condition about the range of $\mu$ to reduce a type 2 confusion and to guarantee the rationality condition.

In general, we cannot reduce all kinds of type 2 confusions by theorem 2. It is important to show the class that can reduce a type 2 confusion. We know some sample example of the class. In the future, we want to derive a necessary and sufficient condition to reduce a type 2 confusion by formulation of the class.

# 4. Application to Cranes Control Problem

## 4.1. Cranes Control Problem




b)

| μ | the time [min] Ave. | S.D. |
|---|---|---|
| 0.0 | 8.98 | 5.46 |
| 0.003 | 4.81 | 2.55 |
| **0.03** | **4.08** | **2.27** |
| 0.3 | 4.82 | 2.67 |
| 1.0 | 5.32 | 3.31 |

Figure 1 : Cranes control problem and its results

**Figure 1. Cranes control problem and its results.**

We apply our theorem to cranes control problem in figure 1a. We consider the cases of two cranes in the coilyard which consists of 100 locations for storing coils. The task is that the crane carries the certain coil from its initial (Start) address to designated (End) address of the coilyard. The couple of the addresses (Start/End) occurs at random when a task has been achieved. Initially, there are two tasks. Each crane operates the most near task. Two cranes run on the same rail to operate for execution of the task. The direct reward is given to the *agent A* who finish its task when *agent A*'s task is finished and the indirect reward is given to the other agent.

Each crane has four situations, that is, $\{ToStart, ToEnd, Vacant, Rolling\}$. *ToStart* is the moving from its initial address to the storage address of the coil. *ToEnd* is the one to put the coil to the designated address. *Vacant* is the one with no allocated task, and *Rolling* is the state that a crane stays at the coil's start/end address, rolls down its string, picks down/up the coil and rolls up its string. We aim to acquire rules for evading collision when agents' situations are ToStart or ToEnd. If the agent is in other situation, agent acts according to the same rules with our previous paper [2]. Our representation of the agent's sensory input is the same with the paper [2]. And there are three basic actions for executing a task such as, {backward-move, forward-move, waiting}.

Initially, we set $W = 3$. If the length of an episode is larger than 3, we set $\mu = 0$ (see section 3.2.2). From equation 4, we should set $\mu < 0.038$.

## 4.2. Results

We show the time [min] to finish a task in Figure 1b. We take the best result when we set $\mu = 0.03$ that is the inside of our theorem. If we set $\mu = 0$, the agent that does not finish any task cannot learn. On the other hand, if we set a large $\mu$, the agent that does not finish any task interferes getting a direct reward. It means that our theorem is very effective to get a reward in such more realistic problem.

# 5. Conclusions

In most multi-agent reinforcement learning systems, reinforcement learning methods for single-agent systems are used. Though it is important to share a reward among all agents in multi-agent

reinforcement learning systems, conventional work has used ad hoc sharing schemes.

In this paper, we focus on the *Rationality Theorem of Profit Sharing* and analyze how to share a reward among all profit sharing agents in the class where there is no type 2 confusion. We show the necessary and sufficient condition to preserve the rationality condition in multi-agent reinforcement learning systems.

If we use this theorem, we can expect to reduce some type 2 confusion without the least desirable situation where expected reward per an action is zero. We have confirmed the effectiveness of our theorem in cranes control problem.

Our future projects include : 1) to derive a necessary and sufficient condition to reduce a type 2 confusion. 2) to extend to other fields of reinforcement learning, and 3) to find efficient real world applications.

## References


[1] Arai, S., Miyazaki, K. and Kobayashi, S., "Generating Cooperative Behavior by Multi-Agent Reinforcement Learning," in *Proceedings of the 6th European Workshop on Learning Robots*, pp.143-157, 1997.

[2] Arai, S., Miyazaki, K. and Kobayashi, S., "Cranes Control Using Multi-agent Reinforcement Learning," in *Proceedings of International Conference on Intelligent Autonomous System 5*, pp.335-342, 1998.

[3] Grefenstette, J. J., "Credit Assignment in Rule Discovery Systems Based on Genetic Algorithms," *Machine Learning. 3*, pp.225-245, 1988.

[4] Miyazaki, K., Yamamura, M. and Kobayashi, S., "On the Rationality of Profit Sharing in Reinforcement Learning," in *Proceedings of the 3rd International Conference on Fuzzy Logic, Neural Nets and Soft Computing*, pp.285-288, 1994.

[5] Miyazaki, K., and Kobayashi, S., "Rationality of Reward Sharing in Multi-agent Reinforcement Learning," in *Proceedings of the 2nd Pacific Rim International Workshop on Multi-Agents Learning*, pp.111-125, 1999.

[6] Watkins, C. J. H., and Dayan, P., "Technical note: Q-learning," *Machine Learning, 8*, pp.55-68, 1992.