Distributed Reinforcement Learning using Bi-directional Decision Making for Multi-criteria Control of Multi-Stage Flow Systems

Kei Aoki, Hajime Kimura, Shigenobu Kobayashi
Tokyo Institute of Technology, 4259 Nagatsuda midori Yokohana Kanagawa Japan

Abstract. Autonomous control systems have been requested recently for large-scale real systems. Distributed reinforcement learning is attracting attention specifically in control of physical flow systems such as lifelines. In this paper, we will introduce a model of Multi-Stage Flow System (MSFS). MSFS is a framework which can describe various physical flow systems. Furthermore, it is effective in handling multi-criteria, multiple constraints under uncertainty and so on that are difficult to solve in conventional methods because of its features. We propose a new bi-directional decision making algorithm based on a least commitment strategy. We apply our method to controlling of real sewerage systems. The simulation results show that only our method satisfies permissible levels and attains the performance within an acceptance level.

1 Introduction

A method of autonomous control has been requested recently for large-scale real system and has been researched extensively. It is requested to lifeline systems such as sewerage systems especially because they are necessary and indispensable to our modern day life.

Real systems have an uncertainty and a time lag, etc., and approaches using Dynamic Programming or Reinforcement Learning (RL) are known[7]. The lifeline systems are of a large-scale and have a wide-area network composed of service centers. Distributed control within the framework of Multi Agent System (MAS) is promising[6]. Thus, approaches that use distributed RL (DRL) are attracting attention. However, the example of applying DRL to the real systems is not common practice. Moreover, the targeted problem class rarely considers multi-criteria and multiple constraints, etc.

We should target a realistic problem class for consideration in the application of the acquired control policy, and clarify features of the real systems that are multi-criteria, multiple constraints, handling of constraints under uncertainty, interactions and so on. Therefore, we model Multi-Stage Flow Systems (MSFS) which can describe various physical flow systems as a new problem class based on these features. However, conventional DRL cannot address these features directly. Therefore, we propose an approach that acquires an appropriate control policy by addressing them directly.

We propose new DRL using Bi-directional Decision Making (BDM) based on the least commitment strategy[8] in order to address multi-criteria of MSFS which treats a smoothing control at the root of MSFS and satisfies constraints of each agent. As opposed to conventional DRL in which each agent makes a decision independently, our method shares decision making among agents bi-directionally by using a tree structure of MSFS. Firstly each agent presents a feasible action set selected in terms of constraint satisfaction, subsequently in the

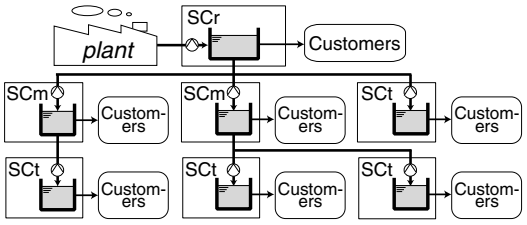

Figure 1: Image of MSFS. Symbol of chevron in circle is a flow controller (e.g. pump).

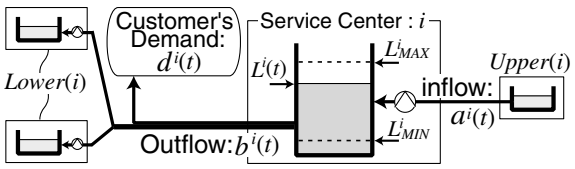

Figure 2: Model of a Service Center in MSFS.

direction of the agent of the root node from the agents of the leaf node. Then, each agent decides unique action in terms of smoothing, subsequently in the direction of the leaf node from the root node. Moreover, we propose a selecting method of the action sets to satisfy the given permissible levels by using value functions to address multi-criteria and multiple constraints. We apply our method to a real sewerage system, and aim to achieve a control policy that fulfills a practicable acceptance level of smoothing while satisfying the given permissible levels of constraint satisfactions.

Conventional methods give the trade-off rates explicitly, and achieve a co-operative behavior within the framework of DRL using a linearly weighted sum (LWS) of value functions [5, 6]. MAXQ[2] hierarchizes a complex task by defining subtasks, and improves the efficiency of learning. However, multiple constraints of MSFS are satisfied by appropriate interactions of agents. In addition, it needs to balance with the purpose of smoothing. Therefore, an appropriate design of weights where such a complex policy is achieved by the approximation of the value function using LWS is complicated. The division of the task into the subtask is also difficult. Our method aims to achieve an appropriate balance not by the trade-off rates that are given by the designer but by BDM and the selection of action sets which satisfy permissible levels. Incidentally, there exists methods using lexicographical optimal policy[3] and policy of handling the risk[4] from the viewpoint of handling constraints our method is similar, though no MAS. In addition, there exists researches which treat distributed constraint satisfaction problems[9], though no RL of control.

In Section 2, we investigate MSFS and how it is modeled and formulated. Section 3 presents our approach and proposal method. Experimental results in a sewerage system show its effectiveness in Section 4. Section 5 concludes this paper.

2 Multi-Stage Flow Systems

2.1 The outline of Multi-Stage Flow Systems

MSFS describe various physical flow systems that target applications such as water supply and sewerage systems, power grids, gas grids, distribution systems and so on.

MSFS is a multi-stage tree structure whose node is the service center (SC) in Fig.1.[1] SCs of the leaf node of tree structure are called terminal SC (SCt), SC of the root is called root SC (SCr) and the rest are called middle SC (SCm). SCr is connected with a special *plant* such as power plants, production plants, treatment plants and so on. Each SC provides service for customers in the newsbeat by controlling the flow received from the upper SC or the *plant*. Flow is a continuous or discrete variable such as water or products. Service is to supply the demand for customers. A main purpose of MSFS is to provide service in just proportion by appropriately controlling flow.

Each SC has to absorb the uncertainty of customer's demand using a buffer such as a

[1] There exists bypasses for emergency, etc. and we exclude them in this paper.

warehouse and a reservoir. However, because a stock level has a bound pair constraint[2]due to limitation of capacity etc., controlling flow that is kept within its range in advance is necessary for a good service.[3] However, it is difficult to completely guarantee the constraint satisfactions under demands with uncertainty. Therefore, a permissible level is set to each SC in general as a permissible rate of the operational constraint violations under the safety margin. A permissible level of each SC is decided beforehand depending on a fluctuation of demand and a capacity of a buffer, etc.

On the other hand, a supply of *plant* to MSFS should be as constant as possible from an economical and operational viewpoint.[4] Therefore, SCs have to absorb the fluctuations of customer's demands by cooperatively controlling flow. And, SCr connected with plant has to control the flow as constant as possible.

Therefore, MSFS is a multi-criteria control problem concerning a smoothing of flow of SCr and multiple constraint satisfactions of stock level of each SC.

2.2 Modelling Multi-Stage Flow Systems

Fig.2 shows the model of SC of MSFS. MSFS consists of N service centers, and the ith SC is described as SC^i (i=0,$\cdots$,$N-1$). $Lower(i)$ denotes the set of SC connected with the lower position of SC^i, and $Upper(i)$ denotes SC connected with the upper position of SC^i. SC^0 is SCr, and $Upper(0)=\emptyset$ because upper facility is a *plant*. SC^i is called SCm when $i>0$ and $Lower(i)\neq\emptyset$. It is called SCt when $Lower(i)=\emptyset$. Each SC^i has a buffer with the bound pair constraint (L^i_{MAX}, L^i_{MIN}).

Each SC^i is characterized by parameters as follows.

- $L^i(t)$ is a stock level of SC^i at t. Below exists as the following constraint conditions.

$$L^i_{MAX} > L^i(t) > L^i_{MIN}. \tag{1}$$

- $d^i(t)$ is the demand of customers of SC^i at t.
- $a^i(t)$ is the inflow that SC^i receives at t, and is a control variable.
- $b^i(t)$ is the outflow of SC^i, and is the sum of the total inflow of $Lower(i)$ and $d^i(t)$.

$$b^i(t) = \textstyle\sum_j a^j(t) + d^i(t), \quad SC^j \in Lower(i). \tag{2}$$

The stock level at $t+1$ is updated by a function $F^i(\cdot)$ which depends on the buffer.

$$L^i(t+1) = L^i(t) + F^i\left(b^i(t) - a^i(t)\right). \tag{3}$$

Negative rewards (penalty) are defined for the control performance as follows.

- $r^i_c\left(L^i(t)\right)$ is penalty of constraint violations where $L^i(t)$ gets out of Eq.1.

[2]Operational constraint is decided by setting a safety margin so as not to arrive at a physical limit in the buffer. Though the system does not break down owing to its violations, the system should be operated keeping it as effective as possible.

[3]For instance in a water supply system, a water level of a pumping plant may arrive at the lower limit if a large demand is generated when the water level is low. It may interfere with providing service. If such a large demand is expected, it is necessary to keep it high by controlling flow in advance.

[4]For instance in a sewerage system, changing an amount of treatment processing is costly because of the biochemistry processing, etc. Thus, smoothing of an inflow is requested of the sewage treatment process. This is similar to the purification process in a water supply system. In a distribution system, smoothing a production volume is requested for reasonable scale of the production line which meets demand under seasonal variations. In a power grid and a gas grid, it is advantageous to suppress the fluctuation of production volume for reasonable scale of plant and a raw procurement.

- $r_s\left(a^0(t), a^0(t-1)\right)$ is the penalty where SCr changes its flow. In regard to flow of SCr, the minimization of a change is equal to the achievement of the smoothing.

Therefore, MSFS is modeled as (N+1) dimension multi-criteria problem.

$$\min_{\boldsymbol{A}} \left\{ \begin{array}{ll} \sum_t r_c^i\left(L^i(t)\right), & i = 0, \cdots, N-1. \\ \sum_t r_s\left(a^0(t), a^0(t-1)\right), & \end{array} \right. \tag{4}$$

where $\boldsymbol{A}$= $(\boldsymbol{a}(0), \cdots, \boldsymbol{a}(t), \cdots)$; $\boldsymbol{a}(t) = \left(a^0(t), \cdots, a^{N-1}(t)\right)$ is a combination of $a^i(t)$ of all SC^i (i=0,$\cdots$,N-1) at all time.

2.3 Formulation to Distributed Reinforcement Learning

MSFS is a distributed system whose each SC^i has the function of a sensor and an actuator as shown by Fig.1 and Section 2.2. The method using the distributed control is considered highly probable in terms of RL[6]. In this paper, MSFS is assumed to be MAS which considers each SC^i to be an agent. We formulate a state-action space and rewards, and approach MSFS with the framework of DRL.

- $a^i(t)$ is an amount of flow and the action of SC^i at t. Flow is a discrete variable, and a continuous value makes it discrete appropriately in this paper.
- $\boldsymbol{s}^i(t)$ is a state of SC^i at t, and is defined as a vector according to the position.

$$\begin{array}{lll} \boldsymbol{s}^0(t) = (T_{ime}(t), L^0(t), \sum_j a^j(t), a^0(t-1)) & ; \ \mathrm{SCr} & , \ SC^j \in Lower(i). \\ \boldsymbol{s}^i(t) = (T_{ime}(t), L^i(t), \sum_j a^j(t)) & ; \ \mathrm{SCm} & , \ SC^j \in Lower(i). \\ \boldsymbol{s}^i(t) = (T_{ime}(t), L^i(t)) & ; \ \mathrm{SCt} & , \ Lower(i) = \emptyset. \end{array} \tag{5}$$

$T_{ime}(t) = t$ (mod τ) is a periodic function at cycle τ concerning t, and denotes the time, the season and so on of an environment. $d^i(t)$ synchronizes at a constant cycle in general, though it cannot be observed beforehand.[5] SCr measures the change of flow (smoothing) by observing $a^0(t-1)$ according to Eq.4. Each SC^i attempts the satisfaction of constraint condition in Eq.1 by observing $L^i(t)$, $\sum_j a^j(t)$ (however at SCt, $Lower(i) = \emptyset$) and $T_{ime}(t)$ in place of $d^i(t)$ according to Eq.2, 3.

- $r^i(t)$ is a reward of SC^i at t, and is defined as $\boldsymbol{r}^0$ is a vector and the rest are a scalar according to Eq.4.

$$\begin{array}{lll} \boldsymbol{r}^0(t) = (r_c^0(L^0(t)), r_s(a^0(t), a^0(t-1))) & ; & \mathrm{SCr}. \\ r^i(t) = r_c^i(L^i(t)) & ; & \mathrm{SCm}, \mathrm{SCt}. \end{array} \tag{6}$$

- $Q^i(\boldsymbol{s}^i(t), a^i(t))$ is a value function corresponding to $r^i(t)$, and denotes the total discounted expected reward based on a certain policy. The value function of SCr is defined as a vector as well as $\boldsymbol{r}^0(t)$. It is calculated by an update rule enhanced based on the SARSA learning [7] (cf. Section 3.5).

$$\begin{array}{lll} \boldsymbol{Q}^0(t) = (Q_c^0(\boldsymbol{s}^0(t), a^0(t)), Q_s(\boldsymbol{s}^0(t), a^0(t))) & ; & \mathrm{SCr}. \\ Q^i(t) = Q_c^i(\boldsymbol{s}^i(t), a^i(t)) & ; & \mathrm{SCm}, \mathrm{SCt}. \end{array} \tag{7}$$

- $p_v^i(t)$ is a constraint violations rate during $[t-\Delta t, t]$. Δt is an arbitrary fixed period.
- p_p^i is a permissible constraint violations rate during Δt. It is called a permissible level, and is given beforehand according to the safety management, etc.

We aim to acquire the control policy of MSFS which achieves a smoothing of the flow of SCr as effective as possible satisfying $p_v^i(t) \leq p_p^i$ under the formulation above.

[5] For instance, customer's necessities that include electricity and water supply increase in the mornings and evenings, then decrease at midnight because of the customer's life cycle.

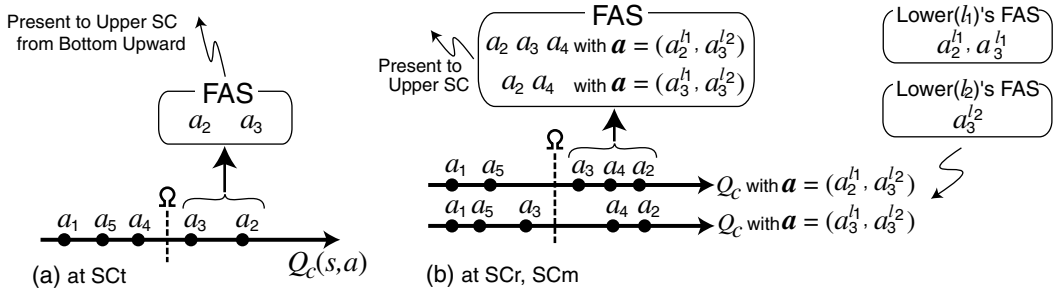

Figure 3: Bottom-up FAS selection. a_k^i is a kth action of SC^i. If $Lower(i)\neq\emptyset$, Q^i is represented with reference to the combination of the lower SC's actions $\boldsymbol{a}$; $Lower(i)=\{l_1, l_2\}$ in (b).

3 Bi-directional Decision Making

3.1 Our Approach

A new method is needed to address MSFS appropriately because of multi-criteria, multiple constraint and so on shown in Section 2. The following points conflict by dividing the state-action space and the rewards into each SC^i as shown in Section 2.3.

- It is necessary to select actions of $Lower(i)$ for selecting action of SC^i beforehand from the definition of $\boldsymbol{s}^i(t)$ in view of the constraint satisfaction. Thus, the decision making process is requested to perform from bottom upwards from SCt to SCr.
- A penalty for smoothing is given to SCr from the definition of $r_s\left(a^0(t), a^0(t-1)\right)$. When SCr selects the action which leads to smoothing, to that end, it is necessary that $Lower(i)$ select accommodative actions. Thus, the decision making process is requested to perform from the top downwards from SCr to SCt.

To settle this competition, we introduce Least Commitment Strategy (LCS) based on the algorithm that Waltz used to understand line drawings[8]. When existing information can be used for decision making locally, this strategy sets aside a part of the decision by only deciding the partial and minimal solution without deciding completely. Then, a complete solution can be obtained based on all partial solutions and interactions.

Our approach based on LCS is as follows. If the actions of $Lower(i)$ are decided, each SC^i can envisage the transition of $L^i(t)$ according to Eq.3 and Eq.5 independently of other SC's. However, each SC^i selects the set of actions estimated to satisfy $p_v^i(t) \leq p_p^i$ according to Q_c^i without deciding the action uniquely as described below, because it cannot address a smoothing at this stage. This set is called Feasible Action Set (FAS$^i(t)$), and FAS of $Upper(i)$ is selected subsequently by presenting FAS$^i(t)$ to $Upper(i)$. After all FAS$^i(t)$ are selected, SCr uniquely selects $a^0(t)$ from FAS$^0(t)$ in view of a smoothing. Decision making is propagated to $Lower(i)$ subsequently, and $a^i(t)$ of each SC^i is decided uniquely. Because each $a^i(t)$ is chosen from FAS$^i(t)$, the satisfaction of $p_v^i(t) \leq p_p^i$ can be expected.

Our approach uses Q_c^i to select FAS$^i(t)$ appropriately. We introduce the threshold $\Omega^i(t)$ of Q_c^i of each SC^i, and choose the action sets with a high possibility to satisfy the permissible levels by adjusting these thresholds. $\Omega^i(t)$ is used to classify $a^i(t)$ in terms of whether $a^i(t)$ in $\boldsymbol{s}^i(t)$ satisfies $p_v^i(t) \leq p_p^i$ from the viewpoint of $Q_c^i(\boldsymbol{s}^i(t), a^i(t))$. Each SC^i updates Q^i according to $r^i(t)$, and improves the control policy.

Therefore, the bi-directional action selection based on LCS and the selection of FAS which uses Ω are the main ideas of BDM. The proposed method consists of the following 4 steps (cf. Section 3.2-3.5), and their steps are performed repeatedly in its algorithm.

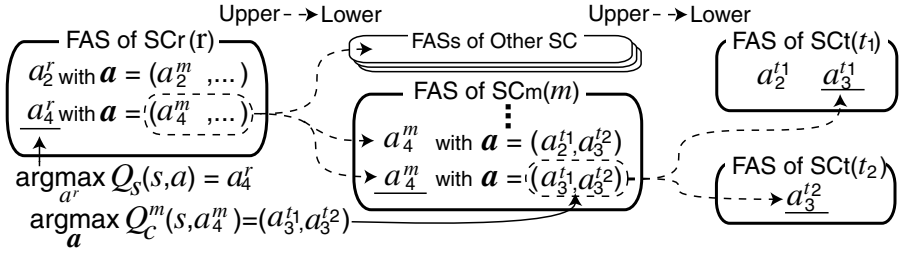

Figure 4: Top-down action selection using FAS. Underlined parts are selected actions. Boxes of broken lines show combinations of actions of the lower SCs related to the selected action. Instruction of actions is indicated by broken line arrows.

3.2 Bottom-up selection of Feasible Action Sets

FAS$^i(t)$ is selected according to Q_c^i concerning the constraints along the following lines. This step performs subsequently from SCt to SCr from the bottom upwards.

- SC^i observes state $\boldsymbol{s}^i(t)$.
- Because of $Lower(i) = \emptyset$ at SCt, $a^i(t)$ are sorted according to $Q_c^i(\boldsymbol{s}^i(t), a^i(t))$ and $a^i(t)$ which satisfy $\Omega^i(t) < Q_c^i$ are added to FAS$^i(t)$ as shown in Fig.3(a).
- Because SCm and SCr have SC$^j \in Lower(i)$, $\sum_j a^j(t)$ is calculable according to a combination $\boldsymbol{a}= (a^j(t)) \, ; a^j(t) \in FAS^j(t)$ that consists of actions that are taken out one by one among each FAS$^j(t)$ that SC^j presents. Thus, the variation of $\boldsymbol{s}^i(t)$ is recognized, and Q_c^i corresponding to each $\boldsymbol{a}$ is addressed as shown in Fig.3(b). Then, $a^i(t)$ which satisfy $\Omega^i(t) < Q_c^i$ as well as SCt are added to FAS$^i(t)$. And, each $\boldsymbol{a}$ is also memorized because Q_c^i depends on $\boldsymbol{a}$.
- When FAS$^i(t)$ becomes an empty set because all actions are excluded by $\Omega^i(t)$, $\mathrm{argmax}_a Q_c^i$ is added to FAS$^i(t)$.
- If $Upper(i) \neq \emptyset$, FAS$^i(t)$ is presented in $Upper(i)$, otherwise this step ends.

3.3 Top-down Selection of Actions

After all SC^i decides FAS$^i(t)$, all actions are decided uniquely subsequently from SCr for smoothing. SCr selects $\mathrm{argmax}_{a^0(t)} Q_s(s^0(t), a^0(t))$ from FAS$^0(t)$ first (a left of Fig.4). Because this $a^0(t)$ accompanies the actions of $Lower(0)$, these actions are instructed to $Lower(0)$ as the broken line shows. Then, SCm (i=m) decides $a^m(t)$ according to $Upper(m)$'s instruction, and gives $Lower(m)$ instructions of their actions similarly. If the action $a^m(t)$ instructed from $Upper(m)$ is a plural in FAS$^m(t)$, the actions of $Lower(m)$ which maximize $Q_c^m(\boldsymbol{s}^m(t), a^m(t))$ are instructed (a center of Fig.4). The action of SCt is decided according to the given instruction (a right of Fig.4).

Therefore, in SCm and SCt which do not obtain a reward of smoothing directly, it is possible to contribute to smoothing as effective as possible satisfying a permissible level. After selecting all actions, each SC^i performs the action concurrently.

3.4 Threshold Adjustment

It is necessary to adjust the threshold $\Omega^i(t)$ appropriately to select the action set to which the FAS selection satisfies a permissible level p_p^i. In this paper, to become $p_v^i(t) = p_p^i$ by comparing an obtained $p_v^i(t)$ with a given p_p^i, $\Omega^i(t)$ is adjusted by a slight value ϵ. Therefore, because the constraint satisfaction has not been achieved then the $p_v^i(t) > p_p^i$, $\Omega^i(t)$ is severely

adjusted. Conversely, because the constraint satisfaction has been achieved then the $p_v^i(t) < p_p^i$, $\Omega^i(t)$ is eased to contribute to smoothing.

$$\Omega^i(t+1) = \begin{cases} \Omega^i(t) + \epsilon & , \quad p_v^i(t) > p_p^i, \\ \Omega^i(t) - \epsilon & , \quad p_v^i(t) < p_p^i. \end{cases} \tag{8}$$

Therefore, $\Omega^i(t)$ is adjusted to the maximum value that $p_v^i(t)$ is suppressed to p_p^i. All $a^i(t)$ that are expected to satisfy $p_v^i(t) \leq p_p^i$ in view of Q_c^i are added to FAS$^i(t)$.

3.5 Updating Value Functions

The SARSA learning (SARSA)[7] is a method for learning the value function of state-action pairs according to trial and error based on a certain policy. In this paper, to update our value functions, we enhance SARSA.

- The update of $\boldsymbol{Q}^0$ of SCr uses SARSA in each (Q_s, Q_c^0). Below, o is s or c.

$$Q_o^0(s(t), a(t)) = (1-\alpha) Q_o^0(s(t), a(t)) + \alpha \left(r_o + \gamma Q_o^0(\boldsymbol{s}(t+1), a(t+1)) \right). \tag{9}$$

- To update the value function Q^i at SC^i; $i > 0$, we enhance SARSA as follows in view of using FAS$^i(t)$ for the decision making.

$$Q^i(s(t), a(t)) = (1-\alpha) Q^i(s(t), a(t)) + \alpha(r + \gamma V_f^i). \tag{10}$$

$$V_f^i = \frac{1}{|\mathrm{FAS}^i(t+1)|} \sum_{a(t+1)'} Q(\boldsymbol{s}(t+1), a(t+1)'), \tag{11}$$

where $a(t+1)' \in \mathrm{FAS}^i(t+1)$ and $|\mathrm{FAS}^i(t+1)|$ is the number of $a(t+1)'$.

Update rule (Eq.9) of SARSA is based on value function $Q^i(\boldsymbol{s}^i(t+1), a^i(t+1))$ of action $a^i(t+1)$ selected at $t+1$. Update rule (Eq.10) of our method is based on value function V_f^i of action set FAS$^i(t+1)$ selected at $t+1$. $Upper(i)$ can give SC^i the instruction of an arbitrary action from among FAS^i as stated above. Considering this arbitrariness, the value function V_f^i of FAS$^i(t+1)$ should reflect the value function $Q^i(\boldsymbol{s}(t+1), a(t+1)')$ of the action $a(t+1)'$ that is included in FAS$^i(t+1)$. In this paper, we define the mean value as the value function V_f^i of FAS$^i(t+1)$ (Eq.11). Therefore, our update rule (Eq.10) which uses FAS is an appropriate extension of the update rule of SARSA (Eq.9).

4 Application to Sewerage System

We take up the control problem of a sewerage system, and confirm utility and availability by the comparison with conventional methods. The target that consists of 5 SC as shown in Fig.5 is modeled on a real sewerage system with large-scale processing in Kawasaki, Japan. The system supplies service of collecting sewage from customers, carrying it through the pump plants, and purifying it in the treatment process. Because of collecting flow, we can model it as MSFS by handling a reverse flow in Fig.1 and Fig.2. The flow $a^i(t)$ can be controlled in about 10 discrete levels by switching several pumps with hourly decisions. The demand $d^i(t)$ for sewage disposal is uncertain so we input data of real demands of about one month repeatedly. Total of the permissible level $\sum_i p_p^i$ is given by 0.015 that is total of the p_p^i of each SC^i at $\Delta t = 24$.

The performance evaluation index of the control policy is defined as the number of pump switches Sw of SCr. It can be considered that the fewer Sw is, the more smoothing will be achieved, because Sw shows the frequency per day in which the amount of processing is changed. The applicable standard that is clarified by the sewerage experts is $Sw \leq 2$ in this real system. This target performance is called an *Acceptance Level*.

4.1 Design of Compared Methods

To confirm the effectiveness of the proposal method, we choose the method of Schneider [6] and Guestrin [5] as comparison methods. To be able to apply these methods, a part of the formulation is changed as follows.

Because the method of Schneider cannot treat the reward of the vector directly, the reward of SCr of Eq.6 is linearly summed with weight β^0. And, it becomes single-criteria. $r^i; i = 1, \cdots, N-1$ of SCm and SCt are the same as Eq.6.

$$\begin{array}{ll} r^0(t) = r_c^0\left(L^0(t)\right) + \beta^0 r_s\left(a^0(t), a^0(t-1)\right) & ; \ \mathrm{SCr.} \\ r^i(t) = r_c^i\left(L^i(t)\right) & ; \ \mathrm{SCm, SCt.} \end{array} \tag{12}$$

The following methods are explained based on [6] under Eq.12 above. In these methods, each SC^i independently decides the action.

- The Local Reward DRL (LRDRL) setting: Each SC^i independently decides the action according to the local reward shown in Eq.12.
- The Global Reward DRL (GRDRL) like setting: All SC^i share the reward r_s concerning a smoothing of SCr.

$$r^i(t) = r_c^i\left(L^i(t)\right) + \beta^i r_s\left(a^0(t), a^0(t-1)\right) \quad ; \quad i = 0, \cdots, N-1. \tag{13}$$

- The Distributed Reward Function (DRF) setting: The reward is shared with connected SC^j. $r^j(t)$ denotes the reward in Eq.12.

$$r^i(t) = \textstyle\sum_j \beta^j r^j(t) \quad ; \quad SC^j \in \mathrm{SC}^i \cup Upper(i) \cup Lower(i), \quad i = 0, \cdots, N-1. \tag{14}$$

- The Distributed Value Function (DVF) setting: The value functions are shared with connected SC^j by using the following update rule under Eq.12. (refer to [6], in detail.)

$$Q^i(s,a) = (1-\alpha)Q^i(s,a) + \alpha\left(r^i + \gamma \textstyle\sum_j f(i,j) \max_{a_j} Q^j(s_j, a_j)\right). \tag{15}$$

Next, for Coordinated Reinforcement Learning (CRL) of Guestrin[5], we consider the *coordination graph* that is comprised of connected relationships which can be seen in Fig.5. The pair of SC^i and SC^j is described as SC^{ij}, the system is addressed by (i,j)={(0,1), (0,2), (0,3), (1,4)}, and we formulate it as follows.

- Though the state is given as follows according to Eq.5, $\sum_j a^j(t)$ is excepted because SC^{ij} contains the interaction.

$$\begin{array}{lll} \boldsymbol{s}^{0i}(t) = \left(T_{ime}(t), L^0(t), L^i(t), a^0(t-1)\right) & ; \ \mathrm{SC}^{0i} & , \quad i = 1, 2, 3. \\ \boldsymbol{s}^{14}(t) = \left(T_{ime}(t), L^1(t), L^4(t)\right) & ; \ \mathrm{SC}^{14}. & \end{array} \tag{16}$$

- The action $\boldsymbol{a}^{ij}(t)$=$(a^i(t), a^j(t))$ is a combination of the flow of SC^i and SC^j. The number of actions becomes the min of 48 and the max of 88 in this system.
- The rewards are defined by a linearly weighted sum as follows according to Eq.6.

$$\begin{array}{ll} r^{0i}(t) = \beta^{0i} r_c^0\left(L^0(t)\right) + \beta^i r_c^i\left(L^i(t)\right) + r_s\left(a^0(t), a^0(t-1)\right) & ; \ \mathrm{SC}^{0i}, i = 1, 2, 3. \\ r^{14}(t) = \beta^{14} r_c^1\left(L^1(t)\right) + \beta^4 r_c^4\left(L^4(t)\right) & ; \ \mathrm{SC}^{14}. \end{array} \tag{17}$$

- The global value function Q is approximated by the sum of local Q^{ij} correspond to r^{ij}. The CRL selects the combination of the actions which maximizes $Q(t)$.

$$Q(t) = \sum_{(i,j)} Q^{ij}\left(\boldsymbol{s}^{ij}(t), \boldsymbol{a}^{ij}(t)\right) \quad ; \quad (i,j) = \{(0,1), (0,2), (0,3), (1,4)\}. \tag{18}$$

Table 1: Performance of the pump switches (Sw) and the constraint violations ($\sum_i p_v^i(t)$).

Method	BDM	LRDRL	GRDRL	DRF	DVF	CRL	$\sum_i p_p^i$
Sw	**1.848**	5.266	3.293	3.593	2.598	3.459	–
$\sum_i p_v^i(t)$	**0.0144**	0.0341	0.0154	0.0307	0.0230	0.0552	0.015

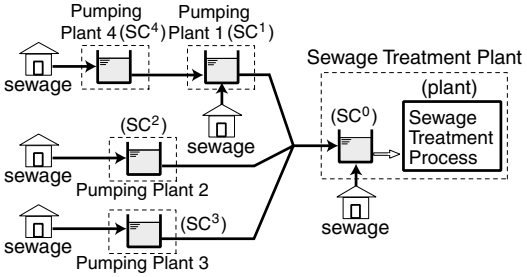

Figure 5: 5 plant Sewerage System.

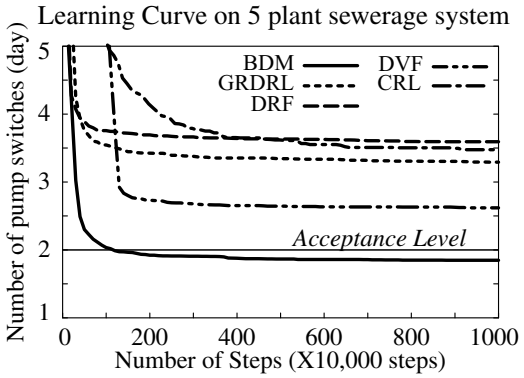

Figure 6: The learning curve on a 5 plant sewerage system. Each is an average of 10 trials. LRDRL is not drawn because its performance is worse than $Sw = 5$.

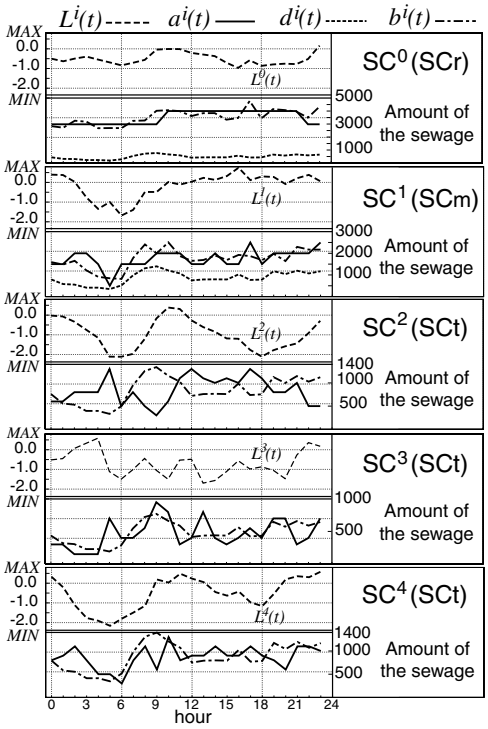

Figure 7: Behaviors of a sewerage system for 24hours based on an average day.

4.2 Experimental Results and Discussion

The learning curve is shown in Fig.6. The plots are an average performance of 10000 steps of 10 trials, and are an average of the best value up to the step. The daily average performances of the control of each method are shown in Table 1. Each weight of the compared methods ($\beta, f(i,j)$) is appropriately set in consideration of permissible levels.

LRGRL hardly achieves smoothing as seen in Sw=5.266. It is suggested that the cooperation of the SCs is indispensable in smoothing the flow of SCr and that it may not be acquired by pursuing a local reward. Though the system using GRDRL and DRF behaves more reasonably than LRGRL that only uses local rewards, neither enough smoothing is achieved. It is suggested that cooperated behavior necessary for smoothing may not be appropriately acquired because the interactions cannot be controlled directly. In GRDRL, the constraint violation as well as Sw is less than LRDRL. Pumping plants stabilize the amount of inflow into SCr by the smoothing reward, and it is suggested that it mitigates the violations of SCr.

Though DVF acquires a tolerable performance, it does not arrive at the performance target. The difficulty in DVF which depends on the linear weighted sum is the selection of a risky action to contribute to smoothing. It is suggested that such a selection may not be achieved according to the fixed trade-off rates.

Though CRL that acquires better performance than DVF has been reported, the target performance is not acquired here. Because the state-action space of SC^{ij} is large, it is considered that the learning steps are insufficient. Therefore, with the state-action space where each SC^i is as large as this real sewerage system, it is suggested that the distributed control of each

SC^i is effective. Moreover, the performance when CRL learns until 100M steps is $Sw = 3.3$, $\sum_i p_p^i = 0.0216$. Thus, it is suggested that approximation of the global value function to obtain expected behaviors is difficult.

The proposal method (BDM) is an only method that acquires the control policy which satisfies an *Acceptance Level*. Though the permissible levels might not be satisfied at the opening of learning because the FAS selection has not enough accuracy, they are gradually satisfied by adjusting the thresholds along with a learning process.

Fig.7 shows the behaviors of a certain 24 hours of the control policy provided by the proposal method. Each SC^i keeps $L^i(t)$ within the range of the bound pair constraints by appropriately controlling $a^i(t)$. On the other hand, $a^0(t)$ is switched only twice at 0900 and 2100 hours in SCr, that is, a smoothing is achieved. The reason is that $L^0(t)$ is stabilized by adjusting $b^0(t)$ giving $Lower(i)$ instructions of the actions appropriately along $a^0(t)$. Especially, in the behaviors at 0500 hours, co-operated control of SC^2 and SC^3 counterbalances the low flow of SC^1 selected so that SC^1 may keep its water level.

5 Conclusions

We showed and modeled MSFS in relation to various physical flow systems which suited real problems. Our proposed model of MSFS is effective in handling the multi-criteria and the multiple constraints that are difficult to solve in conventional methods. We proposed the new DRL using the BDM which can address these features directly for MSFS.

We applied the BDM to the control problem of the sewerage system, and confirmed it fulfilled an acceptance level satisfying permissible levels. Though the BDM is fundamentally applicable to arbitrary MSFS, it becomes inefficient in cases where a certain SC^i connects a lot of SC$^{j \in Lower(i)}$. The connection of about 5 SC per one node seems limited in this sewerage system though it depends on the number of actions of each SC.

Because the value function is calculated with FAS, the optimum of BDM is unexplained. Future works will show theoretical analysis of rationality of BDM. Moreover, we will show the generality of our method by applying it to other applications.

References

[1] K. Aoki, H. Kimura, A. Nagaiwa and S. Kobayashi. Adaptive control of Sewerage Systems using Distributed Reinforcement Learning, *IEE Japan* 123-D-4, 462–469, 2003, (in Japanese).

[2] T. Dietterich. The MAXQ Method for Hierarchical Reinforcement Learning, *Proceedings of the 15th International Conference on Machine Learning*, 118–126, 1998.

[3] Z. Gabor, Z. Kalmar and C. Szepesvari. Multi-criteria Reinforcement Learning, *Proceedings of the 15th International Conference on Machine Learning*, 197–205, 1998.

[4] P. Geibel. Reinforcement Learning with Bounded Risk, *Proceedings of the 18th International Conference on Machine Learning*, 162–169, 2001.

[5] C. Guestrin, M. Lagoudakis and R. Parr. Coordinated Reinforcement Learning, *Proceedings of 19th International Conference on Machine Learning*, 227–234, 2002.

[6] J. G. Schneider, W. K. Wong, A. Moore and M. Riedmiller. Distributed Value Functions, *Proceedings of the 16th International Conference on Machine Learning*, 371–378, 1999.

[7] R.S.Sutton, and A.Barto. Reinforcement Learning: An Introduction, *A Bradford Book*, The MIT Press,1998.

[8] D. Waltz. Understanding Line Drawings of Scenes with Shadows in The psychology of Computer Vision, *McGraw-Hill Book*, 1975.

[9] M. Yokoo, K. Hirayama. Distributed Constraint Satisfaction Algorithm for Complex Local Problems, *Journals of The Japanese Society for Artificial Intelligence*, Vol.15, No.2, pp.348–354, 2000, (in Japanese).