Paper:

Reinforcement Learning Approach for Adaptive Negotiation-Rules Acquisition in AGV Transportation Systems

Masato Nagayoshi*, Simon J. H. Elderton*, Kazutoshi Sakakibara and Hisashi Tamaki*****

*Niigata College of Nursing
240 Shinnan-cho, Joetsu, Niigata 943-0147, Japan
E-mail:{nagayosi, elderton}@niigata-cn.ac.jp
**Toyama Prefectural University
5180 Kurokawa, Imizu, Toyama 939-0398, Japan
E-mail: sakakibara@pu-toyama.ac.jp
***Kobe University
1-1 Rokkodai-cho, Nada-ku, Kobe, Hyogo 657-8501, Japan
E-mail: tamaki@al.cs.kobe-u.ac.jp
[Received March 3, 2017; accepted August 4, 2017]

In this paper, we introduce an autonomous decentralized method for directing multiple automated guided vehicles (AGVs) in response to uncertain delivery requests. The transportation route plans of AGVs are expected to minimize the transportation time while preventing collisions between the AGVs in the system. In this method, each AGV as an agent computes its transportation route by referring to the static path information. If potential collisions are detected, one of the two agents chosen by a negotiation-rule modifies its route plan. Here, we propose a reinforcement learning approach for improving the negotiation-rules. Then, we confirm the effectiveness of the proposed approach based on the results of computational experiments.

Keywords: reinforcement learning, AGV transportation system, negotiation-rules, state space filter, state space construction

1. Introduction

Recently, the problems of planning and operation have been recognized as among the most important issues in production and logistic systems. In particular, transportation planning using automated guided vehicles (AGVs) in steel production, semiconductor production, and warehousing systems has been widely studied from both the theoretical and practical viewpoints [1, 2]. This paper focuses on the AGV routing problem [2]. In this problem, all of the transportation requests are given, but it is necessary to determine a transportation route plan that prevents collisions between AGVs.

An efficient method for responding to ad hoc requests is proposed based on an autonomous decentralized method for the AGV routing problem [3]. First, each AGV, as an agent, finds the shortest route that satisfies the requests assigned to it. If potential collisions (collisions that have been predicted to occur if no pre-emptive action is taken) are detected, one of the two AGVs, as selected by a negotiation-rule, modifies its route. A set of negotiation-rules is used for every collision avoidance action. These rules consist of a condition-part and an action-part. The rule that matches the conditions of the two agents involved in a potential collision is selected from a set of rules.

Here, we propose a reinforcement learning (RL) [4] approach for improving the negotiation-rules. However, it is difficult to construct a state space in the AGV route planning problem. Thus, we introduce a state space filter for adaptive state space construction [5] that (i) will not require specific RL methods and (ii) enables an easier visualization of the filter than the other state space construction methods [6–8]. Furthermore, we confirm the effectiveness of the proposed approach based on the results of three computational experiments.

Several collision-free AGV routing strategies have been proposed [9, 10]. These methods give good performances, but require the determination of the "priority" of each of the AGVs at every node and arc using the routes of all the AGVs. Therefore, they require a central-computer that calculates the priorities prior to the movement of each AGV. On the other hand, the proposed approach requires a central-computer that maintains and calculates the negotiation-rules when AGVs are moving in real time.

This paper is organized as follows. In Section 2, we describe an AGV routing problem. In Section 3, we introduce an autonomous decentralized planning method [3]. In Section 4, we propose a reinforcement learning (RL) [4] approach for improving the negotiation-rules. In Section 5, we investigate the effectiveness of the proposed approach using the results of three computational experiments. Finally, in Section 6, we give a summary of this paper.

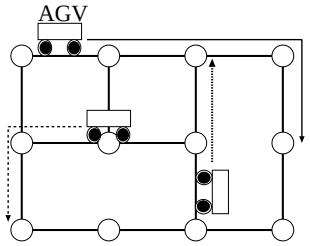


Fig. 1. Example of AGV transportation system.

2. Problem Description

This section describes an AGV route planning problem. An example of our target AGV transportation system is illustrated in **Fig. 1**. There is a set of transportation requests, and a fleet of AGVs is available to fulfill these requests. Each AGV has to move to the destination specified by the request assigned to it, on the rail-guided system.

The transportation problem using AGVs can be broken down into three sub-problems:

(a) the problem of designing the rail network,

(b) the problem of assigning the requests to AGVs,

(c) the problem of routing the AGVs to the destination points.

It is difficult to find routing plans without collisions to solve sub-problem (c). This paper discusses sub-problem (c) under the given rail network and the given assignment of transportation requests as an AGV route planning problem because this would be considered a major issue in practical use.

The AGV route planning problem is defined as follows: N^{V} AGVs V_i $(i = 1,\ldots,N^{\mathrm{V}})$ are available in the rail system. The rail system is represented as N^{N} nodes N_j $(j = 1,\ldots,N^{\mathrm{N}})$ and N^{A} arcs A_k $(k = 1,\ldots,N^{\mathrm{A}})$. A node represents the area for the loading and unloading points and branching points of the rail. An arc represents the path for an AGV. V_i moves to destination node D_ℓ, which is specified by request R_ℓ $(\ell = 1,\ldots,N^{\mathrm{R}})$ assigned to V_i. R_ℓ is given to the system at occurrence time t_ℓ^{R}, and R_ℓ is assigned to V_i. These problem parameters can be summarized as follows:

(a) AGV V_i $(i = 1,\ldots,N^{\mathrm{V}})$,

+ Initial node P_i,

(b) Node N_j $(j = 1,\ldots,N^{\mathrm{N}})$,

(c) Arc A_k $(k = 1,\ldots,N^{\mathrm{A}})$,

(d) Request R_ℓ $(\ell = 1,\ldots,N^{\mathrm{R}})$:

+ occurrence time t_ℓ^{R},
+ destination node D_ℓ.

The following constraints should be taken into account in the planning process:

(1) All AGVs must move synchronously.

(2) The velocity of each AGV is constant.

(3) Each AGV can travel on the arc and turn only at a node. The distance between nodes is constant.

(4) Each arc has a width of one AGV. Therefore, two AGVs cannot simultaneously travel between two nodes from opposite directions.

(5) Each of the AGVs has its own destination node that does not overlap with any of the destination nodes of any other AGV.

Various kinds of criteria may be considered for the evaluation of the routing plan. Here, we consider the maximum completion time $C^{\max}$:

$$C^{\max} = \max t_i^{\mathrm{V}}. \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

where t_i^{V} represents the arrival time at destination node D_i assigned to V_i.

In this paper, we assume that the parameters for R_ℓ are available after time t_ℓ^{R}, i.e., a real-time environment. Therefore, an on-line route planning system is required.

3. Autonomous Decentralized Route Planning System

3.1. Algorithm Based on Negotiation-Rules

In this section, we introduce an autonomous decentralized planning method [3], in which each AGV, as an agent, repeats the planning of its own route and exchanging of route information with the other agents until there is no potential collision. The route of each AGV is calculated using Dijkstra's method [11] on the graph representation. If potential collisions are detected, one of the two AGVs, as selected by a negotiation-rule, modifies its route. The entire route planning process is designed as follows:

(i) Initialization: Each AGV is given an ordinal index $(1,2,3,\ldots)$ and calculates the minimum distance route from its start node to the destination node.

(ii) Information Exchange: All of the AGVs within a certain distance d_{F} exchange their route information and indexes with each other.

(iii) Termination: If any potential collisions are detected until a certain time step t_{L} ahead, the process advances to stage (iv), otherwise the current set of routes are output, and the process moves to the next step.

(iv) Re-planning: One of the two AGVs involved in a potential collision is chosen by a negotiation-rule, which is applied to the AGV with the lowest index, and it re-plans its route to avoid the collision. The process reverts to stage (ii).

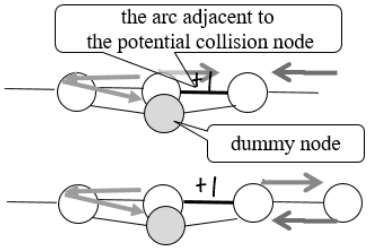


Fig. 2. Virtually adding one dummy node with same arc as node.

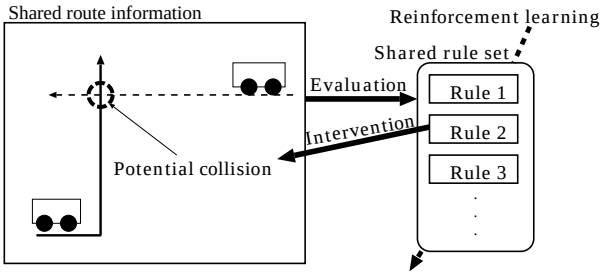


Fig. 3. Framework of rule learning using reinforcement learning.

At stage (iv), a new route plan is acquired by re-planning using Dijkstra's method on the network graph in such a way that the weight value for the arc adjacent to the potential collision node is increased by one.

However, it is impossible to pass through the same node several times using only Dijkstra's method. In this route planning, it is effective to move backward into an adjacent node in order to avoid a collision, thus allowing the other AGV to safely pass without a collision. Thus, it is necessary to pass through the same node several times.

Therefore, we allow the AGV to pass through the same node up to two times using Dijkstra's method by virtually adding one dummy node (with the same arc as the normal node) to each of the nodes, as shown in **Fig. 2**, although it may be effective to pass through the same node more than three times. One of the two AGVs re-creates its route plan in this procedure. In particular, in a case where the AGV is required to re-plan its route, after the weight value for the arc adjacent to the potential collision node is increased by one, Dijkstra's method is used to re-plan the route. Increasing the weight value by one means that the time required for moving is increased by one step. Thus, the AGV's stay at its current node is prolonged by one step. In the case of using a dummy node, if the arc to the dummy node has a weight value of one, then the AGV continues to move to the dummy node. Nishi et al. [2] proposed agent-based AGV routing algorithms using all of the request information in an offline manner prior to execution, where both of the AGVs involved in a potential collision modify their route plans separately. In their approach, both AGVs might move in the same direction to avoid the potential collision node. However, new route plans remain infeasible. In contrast, in our proposed approach, the dummy nodes make it possible to avoid a potential collision. In route planning, it is effective to move backward into an adjacent node in order to avoid a collision, thus allowing the other AGV to safely pass without collision. In some cases this solves the oscillating phenomenon that occurs in the early-stage learning of the negotiation-rules. Our proposed approach is free from such a phenomenon when the negotiation-rules have been sufficiently learned. If the oscillating phenomenon occurs, the value of the negotiation-rule that caused the phenomenon decreases. Thus, the phenomenon ceases to occur as the learning continues. Therefore, the phenomenon does not occur in the later-stage learning.

3.2. Structure of Negotiation-Rules

The AGV that re-plans its own route is determined by a negotiation-rule. In this paper, the negotiation-rule set is shared among all the AGVs.

The negotiation-rule consists of a condition-part and an action-part. The rule that matches the difference between the states of the two AGVs involved in a potential collision is selected from a set of rules. As for the state parameters, we introduce three kinds of variables to represent the whole state space $\boldsymbol{S}$:

s^{O}: The amount of incremental change in the route length after the re-creation,

s^{R}: The route length from the potential collision node to the destination node,

s^{F}: The number of nodes with three or more arcs.

Among the state parameters, s^{F} represents the flexibility of the route creation. Then, $\boldsymbol{S}$ is divided into n^{S} portions $\boldsymbol{S}_u$ $(u = 1, \ldots, n^{\mathrm{S}})$. The action-part expresses whether or not the AGV with the lower index re-creates its route.

The reinforcement learning approach described in Section 4 is applied to acquire a good rule set.

4. Reinforcement Learning Approach for Acquisition of Negotiation-Rules

4.1. Basic Idea

It is difficult to design good negotiation-rules, as shown in Section 3.2, in advance without an advance knowledge of the object system. Here, we propose a reinforcement learning (RL) [4] approach to improve the negotiation-rules. A framework for rule learning using RL is shown in **Fig. 3**. However, this framework can only be formulated as partially observable Markov decision processes (POMDPs), not as Markov decision processes (MDPs). Because this framework does not make it possible to observe the positions of all the AGVs, it is not possible to determine the next state of the negotiation-rule. However, the probability of the next state can be determined through updating the probability distribution of the next state from the current state using RL. Specifically, we use

Q-learning (QL) [12] which is one of the typical RL methods, where state and action spaces are constructed using the condition-parts of the negotiation-rules and the action-parts, namely 0 or 1, respectively. Here, the negotiation-rules consist of the state space of QL.

It is necessary to design a positive reinforcement signal (reward) based on an evaluation of the route plans of all the AGVs. In this paper, the reward is given to a RL agent when all the AGVs arrive at the destination node allowing the maximum completion time $C^{\max}$ to be evaluated. This reward setting is considered to be effective, even if AGVs move asynchronously.

However, it is difficult to construct a state space in the AGV route planning problem.

4.2. State Space Filter for Adaptive State Space Construction

We have proposed a state space filter based on the entropy [5], which is defined by action selection probability distributions in a state, for adaptive state space construction. This state space filter (i) does not require specific RL methods and (ii) enables an easier visualization of the filter than the other state space construction methods [6–8]. Here, the entropy of the action selection probability distributions using Boltzmann selection in a state $H(s)$ is defined by

$$H_{\mathrm{D}}(s) = -\left(\frac{1}{\log|\boldsymbol{A}|}\right)\sum_{a\in\boldsymbol{A}}\pi(a|s)\log\pi(a|s). \quad . \quad (2)$$

where $\pi(a|s)$ specifies the probabilities of taking each action a in each state s, $\boldsymbol{A}$ is the action space and $|\boldsymbol{A}|$ is the number of available actions.

The state space filter is adjusted by treating this entropy $H(s)$ as an index of the correctness of the state aggregation in state s. In particular, in a case where the mapping from the inner state space is roughly digitized to the state space, a perceptual aliasing problem occurs. In other words, the action that an agent should select cannot be clearly identified. Thus, the entropy may not be small in the state space that should be divided. In this paper, the sufficiency of the number of learning opportunities is judged using a threshold value θ_{L}.

Therefore, if the entropy does not become smaller than a threshold value θ_{H}, despite the sufficient number of learning opportunities, the state space filter will be adjusted by dividing the state, as a result of the occurrence of the perceptual aliasing problem. Through this operation, the size of the divided state space increases by (2^3-1), where 3 is the number of dimensions. In addition, please note that the values of the new 2^3 states are those of the state before having been divided.

Similarly, if the entropy is smaller than θ_{H} in both a state s and a different state s', mapping from a transited input state, in addition to the representative actions in the states being the same, then the state space filter will be adjusted by integrating the states as a result of the states being too divided. Moreover, please note that the value of the new state is an average of the values of the two states before being integrated. In addition, if a state s exists that has never been mapped ever once during a certain period θ_{t}, then the state space filter will be adjusted by integrating s and a state adjacent to s. It should also be noted that the values of the new state are those of the state adjacent to s before having been integrated. Through these operations, the size of the state space after being integrated decreases by one. If the size of the state space of the state space filter is small, then the negotiation-rules are easily understood, and it is easy to analyze what variables are necessary. On the other hand, a larger size for the state space requires more memory, and more computational time.

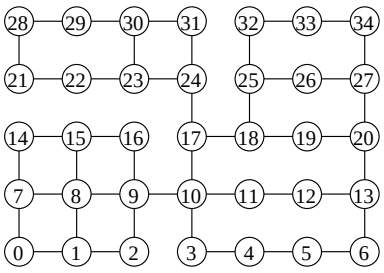


Fig. 4. Rail network of N35.

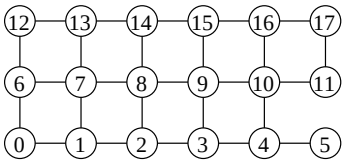


Fig. 5. Rail network of N18.

5. Computational Experiments

5.1. Experimental Settings

The effectiveness of the proposed approach is investigated in this section. We prepared three AGV route planning problems (hereafter called "N35," "N18-1," and "N18-2").

The parameters of N35 are described as follows:

$$\text{N35: } N^{\mathrm{V}}=6, \{\mathrm{P}_i,\mathrm{D}_i\}=\{0,34\},\{1,30\},\{7,6\},\{13,21\},\{28,16\},\{34,0\}.$$

The rail network of N35 is shown in **Fig. 4**. In this rail network, heavy congestion is predicted to occur around nodes 9 and 10.

The parameters of N18-1 and N18-2 are described as follows:

$$\text{N18-1: } N^{\mathrm{V}}=6, \{\mathrm{P}_i,\mathrm{D}_i\}=\{0,17\},\{17,0\},\{6,11\},\{11,6\},\{13,4\},\{4,13\}.$$

$$\text{N18-2: } N^{\mathrm{V}}=6, \{\mathrm{P}_i,(\mathrm{D}_{0i},\mathrm{D}_{1i})\}=\{0,(17,0)\},\{17,(0,17)\},\{6,(11,6)\},\{11,(6,11)\},\{13,(4,13)\},\{4,(13,4)\}.$$

Table 1. Parameters for experiments.

Parameter	Value
α	0.1
γ	0.9
τ	0.1
θ_{H}	0.47
θ_{L}	1000
θ_{t}	100 episodes

The rail network used for both N18-1 and N18-2 is shown in **Fig. 5**. In this rail network, potential collisions occur as often as we can design them, which may be more often than in real rail networks. In N18-2, each AGV makes a round trip from the initial node to the destination node on the rail network of N18. One step is defined as the time step for moving to the adjacent node. One episode is defined as the steps needed to accomplish the task or 100 steps, which ever comes first. In each episode, each AGV departs from its initial node. All of the occurrence times $\forall l, t_l^{\mathrm{R}} = 0$, the detectable time step $t_{\mathrm{L}} = 2$, and the exchangeable distance $d_{\mathrm{F}} = 2$, which is the Manhattan distance. The smallest numbers of steps required for accomplishing N35 and N18-1, assuming none of the AGVs collide with each other, are 10 and 7, respectively.

To apply QL with a state space filter, a three-dimensional initial state space is designed with an initial size of one (hereafter called "SF"). For comparison, three state space constructions with a one-dimensional state space are designed so that the state space is evenly divided into two (hereafter called "2-1-1," "1-2-1," and "1-1-2") to apply QL without a state space filter. In addition, a three-dimensional state space is designed so that the state space is evenly divided into $2 \times 2 \times 2$ (hereafter called "2-2-2") to apply QL without a state space filter.

Further, for a comparison with a conventional method, Maza's method was used [9], as one of the conventional methods proposed with a two-stage approach: one stage for finding the shortest routes (on the condition that all the AGVs could not consider detour routes) for AGVs to determine their priorities at every node and arc using the routes of all the AGVs, and the other stage for avoiding collisions while following the determined shortest routes of the AGVs. However, Maza's method only uses the shortest routes (without considering detour routes). Therefore, when the routes of two AGVs overlap (when the shortest respective routes of two AGVs moving in opposing directions overlap), the resulting routing problem cannot be resolved in N18-1 and N18-2. Therefore, the number of steps required by Maza's method is only investigated for the N35 problem. Here, their priorities are calculated based on the route lengths from the initial to destination nodes of all the AGVs.

The reward $r = 10$ (reward) is given to the agent only when all the AGVs arrive at their destination nodes; the reward $r = -5$ (punishment) is given to the agent only when the weight of the arc is more than 100; and the reward $r = 0$ is given to the agent at any other node.

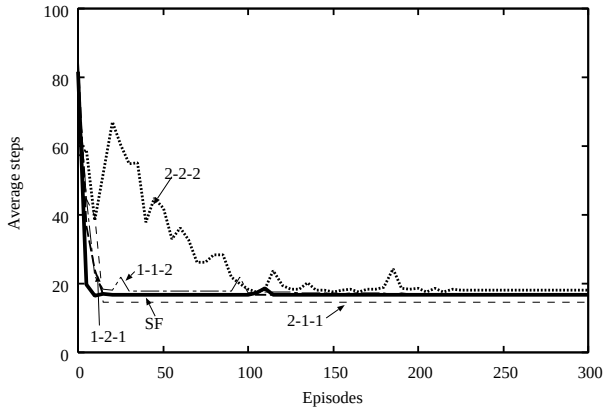


Fig. 6. Required steps for N35.

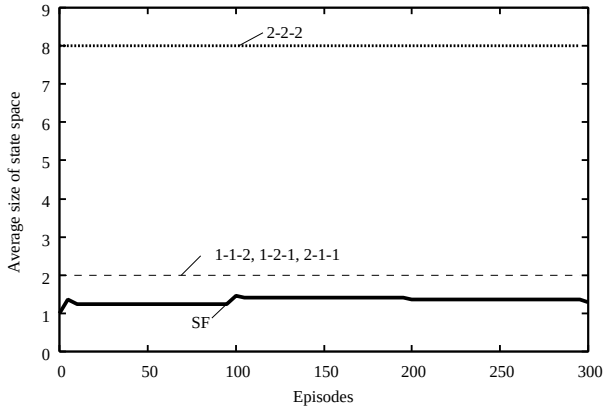


Fig. 7. Size of state space for N35.

Computational experiments were performed using the parameters listed in **Table 1**, where α is the learning rate, γ is the discount factor, and τ is the temperature for Boltzmann selection.

In addition, all the initial Q-values were set at 5.0 as optimistic initial values. Here, θ_{H} was set at approximately 0.47, which was the maximal value of the entropy when the highest selection probability for one action was 0.9 from within two available actions.

5.2. Results

The average number of steps required and the sizes of the state spaces needed to solve N35, N18-1, and N18-2 problems were observed during learning over 20 simulations with different state constructions, as described in **Figs. 6**, **7**, **8**, **9**, **10**, and **11**, respectively.

Cross-sections of the negotiation-rules obtained by SF for N18-1 at 50 episodes and 150 episodes are shown in **Fig. 12**. Here the gray cells show re-routing, and the size of the circle in the cell shows the maximum Q-value in the state. Stages 1 and 2 show cross-sections of the negotiation-rules at episode 50 (taken as an example of early-stage learning) and at episode 150 (taken as an example of late-stage learning). Because the negotiation-rules were composed of three-dimensional variables, stages 1 and 2 show two-dimensional cross-sections from near the center of each space at $s^{\mathrm{O}} = 1$,

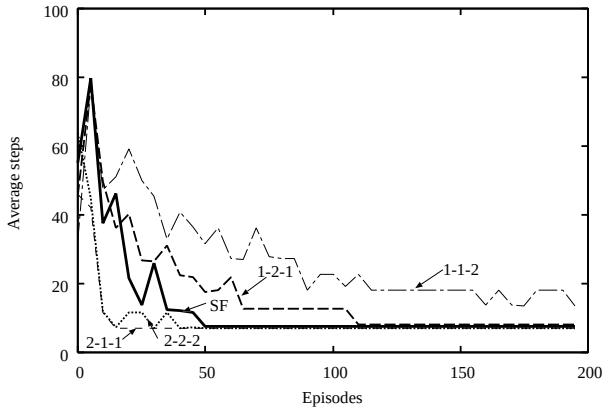


Fig. 8. Required steps for N18-1.

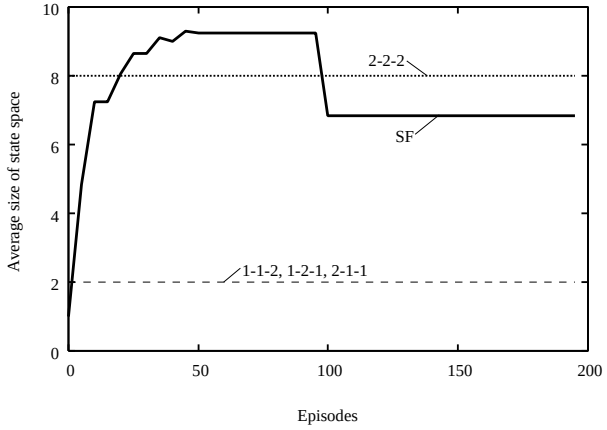


Fig. 9. Size of state space for N18-1.

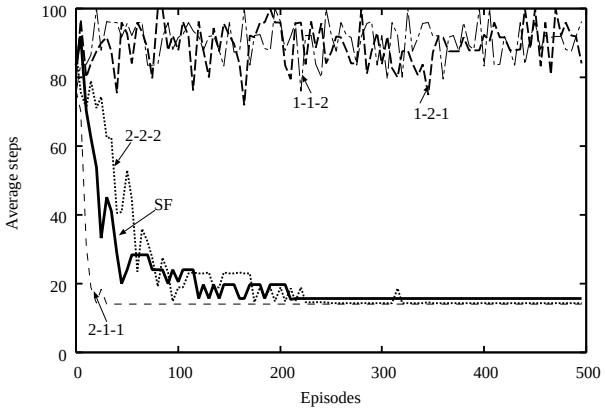


Fig. 10. Required steps for N18-2.

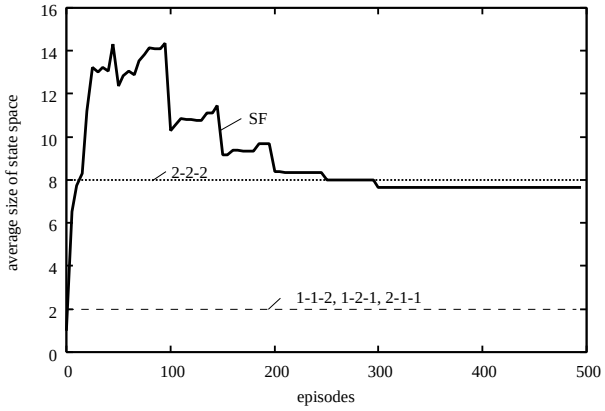


Fig. 11. Size of state space for N18-2.

$s^{\mathrm{R}} = 1$, and $s^{\mathrm{F}} = 1$. Stage 3 shows two-dimensional cross-sections far from the center of each space at $s^{\mathrm{O}} = 8$, $s^{\mathrm{R}} = 8$, and $s^{\mathrm{F}} = 8$.

The variance of the required steps under the 20 simulations by SF and the numbers of steps required by Maza's method for N35, N18-1, and N18-2 listed in **Table 2**.

The average number of steps required and the sizes of the state spaces needed to solve the N18-1 problem were observed during learning over 20 simulations with five initial state constructions, 1-1-1, 1-1-2, 1-2-1, 2-1-1, and 2-2-2, with the state space filter (hereafter called "SF1-1-1," "SF1-1-2," "SF1-2-1," "SF2-1-1," and "SF2-2-2," respectively), as described in **Figs. 13** and **14**, respectively.

Finally, the best routes acquired for N35 and N18-2 by SF are shown in **Figs. 15** and **16**, respectively. The colored circles signify the different AGVs, and the colored triangles signify their destinations. Here, the size of the state space of SF was three when SF acquired the best routes for N35.

The following can be seen in **Figs. 6** and **7**:

1. The proposed approach could acquire feasible route plans that just exceeded the least number of required steps, and could perform well in N35, even if the size of the state space was one.

2. 2-2-2 showed a worse performance than SF, 2-1-1, 1-2-1 and 1-1-2 with regard to the learning speed, because the size of the state space was larger than any other construction.

The following can be seen in **Figs. 8** and **9**:

1. With the exception of 1-1-2, the proposed approaches could acquire feasible route plans just exceeding the least number of required steps.

2. 1-1-2 showed a worse performance than any other construction. Therefore, s^{F} is not important for N18-1.

The following can be seen in **Figs. 10** and **11**:

1. 1-2-1 and 1-1-2 could not acquire any appropriate negotiation-rules. Therefore, no appropriate negotiation-rules could be constructed using either variable s^{R} or s^{F} in N18-2. The results pointed to the conclusion that neither s^{R} nor s^{F} was important.

2. 2-1-1 showed a better performance than any other construction. Therefore, s^{O} is important for N18-2.

The following can be seen in **Fig. 12**:

1. In accordance with the progression of learning, the states were appropriately integrated for the following reasons. **Figs. 8** and **9** confirm that the performance was maintained at episode 150, whereas, at episode 50, the part that was divided too finely in the state space was integrated at episode 100.

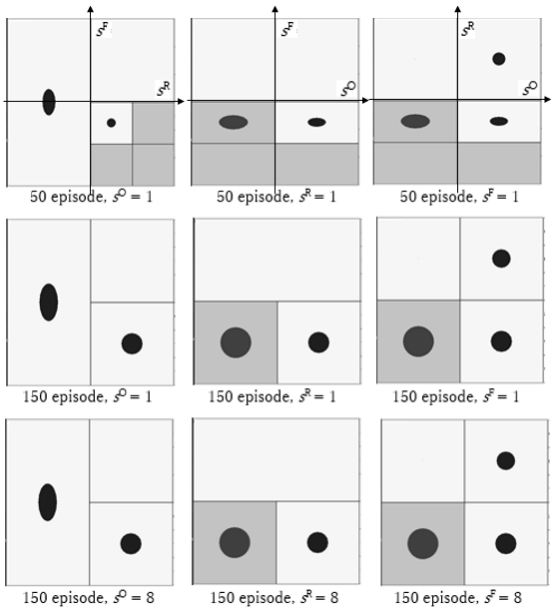


Fig. 12. Cross-sections of the negotiation-rules by SF for N18-1.

Table 2. Variance of required steps by SF and number of steps required by Maza's method for experiments.

Problem	SF		Maza's method
	Average	Standard deviation	
N35	15.6	1.8	15
N18-1	7.7	0.8	-
N18-2	16.5	1.5	-

2. At the 150th episode in N18-1, the cross-sections of the negotiation-rules at $s^{\mathrm{O}} = 1$ and $s^{\mathrm{O}} = 8$; and at $s^{\mathrm{R}} = 1$ and $s^{\mathrm{R}} = 8$; and at $s^{\mathrm{F}} = 1$ and $s^{\mathrm{F}} = 8$ were the same respectively. Thus, whether the values were positive or negative was the only relevant aspect.

3. At episode 150, it can be seen that at the cross-section of $s^{\mathrm{R}} = 1$, we confirmed that the Q-value of the action where the route was re-planned, under the conditions $s^{\mathrm{O}} < 0$ and $s^{\mathrm{F}} < 0$, was larger than the Q-value of the action where the route was not re-planned, under the conditions $s^{\mathrm{O}} \geq 0$ and $s^{\mathrm{F}} < 0$. In other words, we can conclude that at $s^{\mathrm{R}} = 1$, the negotiation-rule where $s^{\mathrm{O}} < 0$ and $s^{\mathrm{F}} < 0$ is closer to being able to acquire the reward than the negotiation-rule where $s^{\mathrm{O}} \geq 0$ and $s^{\mathrm{F}} < 0$, because the Q-values become larger the closer they are to being able to acquire the reward.

The following can be seen in **Table 2**:

1. SF had stability over the three problems.

2. SF showed a slightly worse performance than Maza's method using the routes of all the AGVs in N35. However, SF has the potential of acquiring shorter routes than Maza's method.

The following can be seen in **Figs. 13** and **14**:

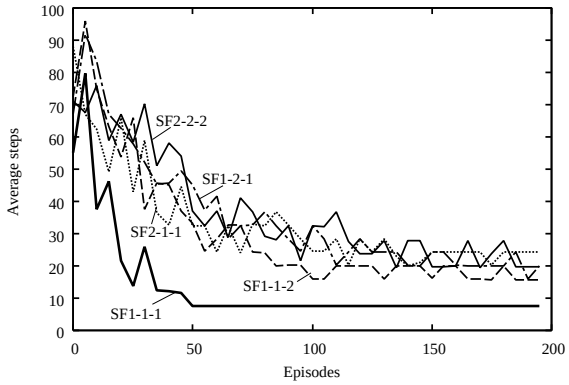


Fig. 13. Required steps for N18-1 by QL with various initial state space filters.

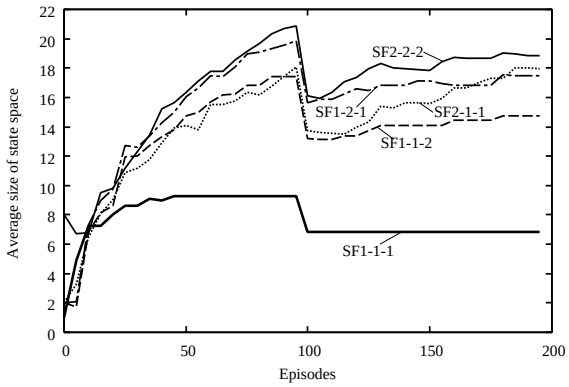


Fig. 14. Size of state space for N18-1 by QL with various initial state space filters.

1. SF1-1-1 had the best performance. This may have been for the following reasons. As the size of the initial state space decreased, it became more connected to the generalization of the reward, and the small size of the initial state space was related to the promotion of the learning speed. On the other hand, the over-generalization of the reward was connected to the perceptual aliasing problem, leading to the inability to learn effectively. Our proposed approach avoided this problem by adjusting the state space filter through dividing the state. As a result, SF1-1-1 had a better performance than the other initial state space filters.

The following can be seen in **Figs. 15** and **16**:

1. The AGVs could acquire appropriate routes to move backward to avoid collisions. **Fig. 15** shows what are considered to be the optimal routes because they are shorter than the routes obtained by Maza's method. **Fig. 16** also shows the optimal routes.

2. It was confirmed that, in step 5 of N35, in order to move backward to node 31 to avoid a collision, the AGV passed through node 24 twice. Similarly, in step 7, in order to move backward to node 25 to avoid a collision, the AGV passed twice through nodes 17 and 18.

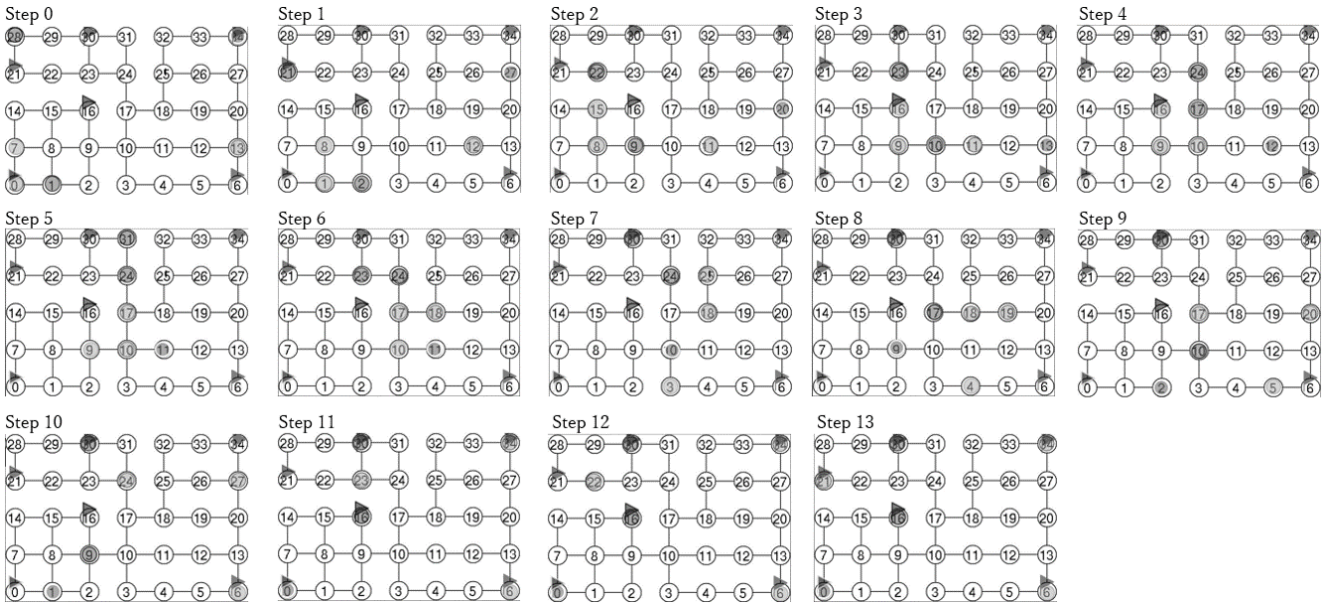


Fig. 15. Best acquired routes by SF for N35.

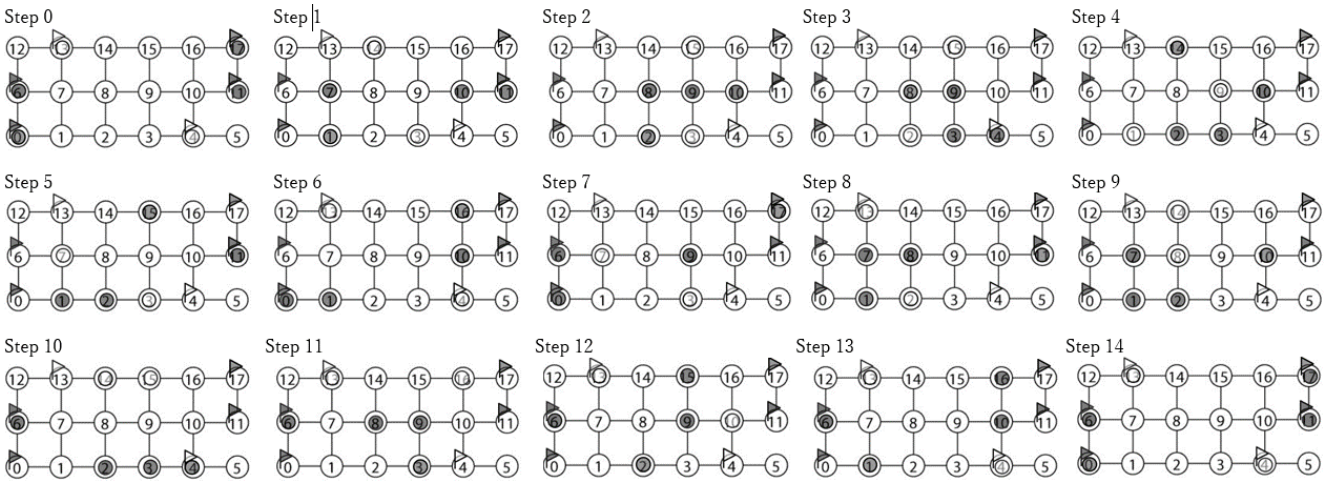


Fig. 16. Best acquired routes by SF for N18-2.

Consequently, we confirmed the following:

1. SF and 2-1-1 showed better performances than any other construction for the three problems.

2. It was difficult to construct a state space.

3. The importance of the variables differed depending on the problems.

4. Therefore, SF could acquire feasible and efficient rule sets for the problems for the following reasons. Although 2-1-1 showed the best performance in the three computational experiments, because the importance of the variables differed depending on the problem, it may not be possible to resolve other problems with 2-1-1, with only uses the variable s^{O}.

6. Conclusions

This paper considered the AGV route planning problem and introduced an autonomous decentralized route planning method. To realize a solution, we introduced an autonomous decentralized route planning method, in which each AGV, as an agent, computes its transportation route by referring to the static path information, and exchanges route plans with the other AGVs. If potential collisions are detected, one of the two agents, as selected by a negotiation-rule, modifies its route plan. Here, we proposed a reinforcement learning approach for improving the negotiation-rules. In three computational experiments, it was observed that when the proposed approach, particularly a state space filter, was applied, feasible and efficient rule sets could be acquired for the problems.

Our future projects include evaluating the effective-

ness of our proposed approach under the condition that more than one request is assigned to each AGV, comparing comprehensively the performance of our proposed approach in rail network settings with wider applicability, and confirming the effectiveness of our proposed approach when there are changes in the network or differences in the topology or size of the network.

References:

[1] Y. Seo and P. J. Egbelu, "Integrated Manufacturing Planning for an AGV-based FMS," Int. J. of Production Economics, Vol.60-61, pp. 473-478, 1999.

[2] T. Nishi, K. Sotobayashi, M. Ando, and M. Konishi, "Evaluation of an Agent-Based Transportation Route Planning Method for LED Fabrication Line using Lagrangian Relaxation Technique," Proc. of Int. Symp. on Scheduling 2002, pp. 71-74, 2002.

[3] K. Sakakibara, Y. Fukui, and I. Nishikawa, "Genetic-Based Machine Learning Approach for Rule Acquistion in an AGV Transportation System," Proc. of ISDA 2008, Vol.3, pp. 115-120, 2008.

[4] R. S. Sutton and A. G. Barto, "Reinforcement Learning," A Bradford Book, MIT Press, 1998.

[5] M. Nagayoshi, H. Murao, and H. Tamaki, "A State Space Filter for Reinforcement Learning," Proc. of AROB 11th'06, pp. 615-618, 2006.

[6] R. S. Sutton, "Generalization in Reinforcement Learning: Successful Examples Using Sparse Coarse Coding," Advances in Neural Information Processing Systems: Proc. of the 1995 Conf., pp. 1038-1044, 1996.

[7] K. Doya, "Temporal Difference Learning in Continuous Time and Space," Advances in Neural Information Processing Systems: Proc. of the 1995 Conf., pp. 1073-1079, MIT Press, 1996.

[8] H. Murao and S. Kitamura, "QLASS: an enhancement of Q-learning to generate state space adaptively," Proc. of Fourth European Conf. on Artificial Life(ECAL97), 1997.

[9] S. Maza and P. Castagna, "A performance-based structural policy for conflict-free routing of bi-directional automated guided vehicles," Computers in Industry, Vol.56, No.7, pp. 719-733, 2005.

[10] A. ter Mors and C. Witteveen, "Plan repair in conflict-free routing,," Lecture Notes in Artificial Intelligence, pp. 46-55, Springer Verlag LNAI, 2009.

[11] E. W. Dijkstra, "A note on two problems in connexion with graphs," Numerische Mathematik, pp. 269-271, 1959.

[12] C. J. C. H. Watkins and P. Dayan, "Technical note: Q-Learning," Machine Learning, Vol.8, pp. 279-292, 1992.

Name:
Masato Nagayoshi

Affiliation:
Associate Professor, Humanities and Environmental Sciences, Niigata College of Nursing

Address:
240 Shinnan-cho, Joetsu 943-0147, Japan

Brief Biographical History:
2007- Instructor, Humanities and Environmental Sciences, Niigata College of Nursing
2008- Assistant Professor, Humanities and Environmental Sciences, Niigata College of Nursing
2013- Associate Professor, Humanities and Environmental Sciences, Niigata College of Nursing

Main Works:
• "An Entropy-Guided Adaptive Co-Construction Method of State and Action Spaces in Reinforcement Learning," Lecture Notes in Computer Science (LNCS), Vol.8834, pp. 119-126, 2014.

Membership in Academic Societies:
• The Institute of Electrical Engineers of Japan (IEEJ)
• The Institute of Systems, Control and Information Engineers
• The Society of Instrument and Control Engineers (SICE)

Name:
Simon J. H. Elderton

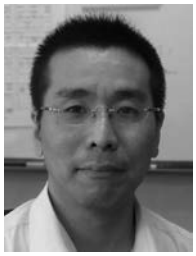
Affiliation:
Senior Lecturer, Humanities and Environmental Sciences, Niigata College of Nursing

Address:
240 Shinnan-cho, Joetsu, Niigata 943-0147, Japan

Brief Biographical History:
1993- Assistant Language Teacher, JET Program
1998- Head of Languages, Nelson College, New Zealand
2008- Assistant Professor (English), Niigata College of Nursing

Main Works:
• "A Literature Review on 22q11.2 Deletion syndrome: The need for patient and family care management in Japan," J. of Japanese Society of Genetic Nursing, Vol.14, No.2, pp. 53-63, 2016.

Membership in Academic Societies:
• Japan Association of Language Teachers
• Japanese Society of Genetic Nursing
• The Japan Association of Comparative Culture



Name:
Kazutoshi Sakakibara

Affiliation:
Associate Professor, Toyama Prefectural University

Address:
5180 Kurokawa, Imizu, Toyama 939-0398, Japan

Brief Biographical History:
2004- Ritsumeikan University
2013- Toyama Prefectural University

Main Works:
- Mathematical programming modeling of production and logistic process
- Optimization algorithm based on mathematical programming
- Agent-based simulation

Membership in Academic Societies:
- The Society of Instrument and Control Engineers
- The Institute of Electrical Engineers of Japan
- The Institute of Systems, Control and Information Engineers

Name:
Hisashi Tamaki

Affiliation:
Professor, Graduate School of System Informatics, Kobe University

Address:
1-1 Rokko-dai, Nada-ku, Kobe 657-8501, Japan

Brief Biographical History:
1990- Research Associate, Faculty of Engineering, Kyoto University
1995- Lecturer, Faculty of Engineering, Kobe University
1999- Associate Professor, Faculty of Engineering, Kobe University
2006- Professor, Faculty of Engineering, Kobe University
2010- Professor, Graduate School of System Informatics, Kobe University

Main Works:
- "Model Structure and Learning Process for a Driver Model Capable to Improve Driving Behavior," J. of Control Engineering and Technology, Vol.3, No.2, pp. 41-49, 2013.

Membership in Academic Societies:
- The Institute of Systems, Control and Information Engineers
- The Institute of Electrical Engineers of Japan
- The Iron and Steel Institute of Japan