Avoidance method for unexpected impassable paths in traffic simulator
RoboCup 2018 – TDP Infrastructure
AIT-Rescue (Japan)

Yuki Miyamoto[1], Kazunori Iwata[2], and Nobuhiro Ito[1]

[1]Department of Information Science, Aichi Institute of Technology, Japan
[2]Department of Business Administration, Aichi University, Japan
[rescue-2018@maslab.aitech.ac.jp]

Abstract. A new traffic simulator that eliminates the unexpected behaviors of the current traffic simulator is proposed. We focused on two problems related to deriving incorrect movement paths. To improve simulation, paths must be driven by considering the shape and size of an agent and the state of connection between roads. The present paper describes these problems and the specifications of the new simulator.

1 Introduction

A traffic simulator is a sub-simulator used in RoboCupRescue Simulation [1]. The traffic simulator reflects the movement commands of all agents and the load/unload commands of Ambulance Teams in the simulation space.

However, the current traffic simulator may transfer an agent along an unexpected path depending on the blockage conditions of a road because the derivation method in the simulator faces problems.

To solve the problems, the method of deriving the path must be improved. This paper therefore develops a new traffic simulator. The new traffic simulator uses a derivation method that considers the properties of agents. In an evaluation, we confirmed that our simulator works well for expected examples. We anticipate that our simulator will meet relevant requirements.

Section 2 explains the features and problems of the current traffic simulator. Section 3 describes the new traffic simulator. Section 4 presents examples of the use of the new simulator.

2 Current Traffic Simulator

2.1 Control of agent movement

Figure 1 is a simplified flow diagram of how to reflect agent movement in the simulation space of RoboCupRescue Simulation, The current traffic simulator firstly receives the movement commands of all agents from the kernel in ①. A movement-command object includes two member variables given in Table 1. The simulator plans a detailed movement path for each agent using those variables in ②. Finally, the simulator transfers all agents while adjusting their coordinates, which give the locations of agents, through collisions among multiple agents and/or collisions between agents and walls/blockages in ③.

This paper refers to a movement path planned by an agent as a sequence of roads and a detailed movement path planned by a simulator as an actual path.

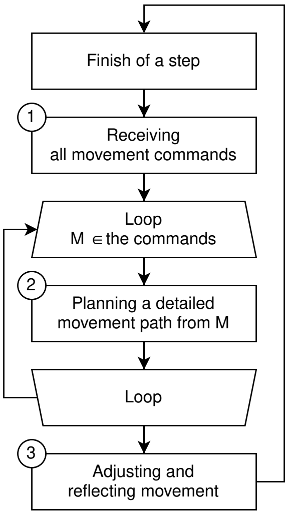

Fig. 1. Reflection from an agent's movement to the simulation space

Table 1. Member variables in a movement command

Member	Overview
Path	Ordered sequence of IDs to the destination planned by the agent
Destination	Coordinates where the agent goes forward

2.2 Planning an actual path for each agent

From the sequence of roads and the destination included in an agent's movement command, the simulator plans an actual path composed from coordinates on the roads.

The simulator derives the actual path that combines those actual paths in order of the sequence of roads after planning the actual path for each road. The simulator employs the following two methods to derive an actual path of a road.

- Method 1. Connecting only three pairs of coordinates on the road
- Method 2. Creating a graph within the road and performing path planning

Only if the actual path of the method 1 collides with walls or blockages does the simulator employ method 2, which can derive an actual path that avoids the blockages. The details of these methods are explained as follows.

Method 1: Connecting only three pairs of coordinates on the road As shown in Fig. 2, this method derives an actual path of the road by connecting the entrance coordinates(① in Fig. 2) and the central(②) and exit coordinates(③). The entrance/exit coordinates are selected from the coordinates that are most distant from blockages on the entrance/exit that lies on the dividing line between adjacent roads.

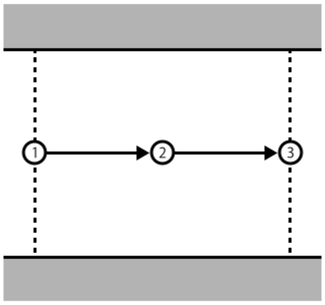

Fig. 2. Actual path obtained using current method 1

Method 2: Creating a graph within the road and performing path planning As shown in Fig. 3, the method creates a graph using the cells produced by a simple decomposition of the road. Moreover, it performs path planning on the graph and obtains an actual path as shown in Fig. 4.

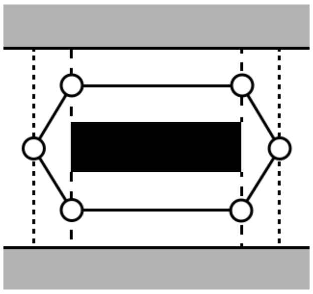

Fig. 3. Graph of current method 2

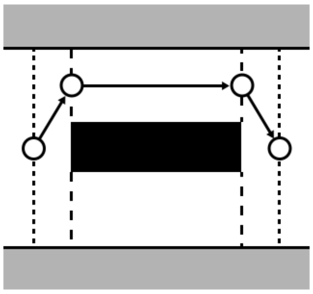

Fig. 4. Actual path obtained using current method 2

2.3 Adjusting actual paths of all agents and reflecting their movement

The simulator transfers all agents along the actual paths planned in section 2.2 while adjusting their movement locations by collisions with agents, walls or blockages. The current simulator also represents traffic congestion by a principle of action and reaction as shown below.

The simulator decides the movability of an agent and the distance through interactions of the following forces [2].

- Force acting toward the next coordinates on the actual path
- Moving velocity defined for agents
- Current actual velocity of agents
- Force received from other agents or walls/blockages

In addition, the simulator adopts a concept of time defined by dividing a step more finely, to make calculations with these forces.

2.4 Problems of the current traffic simulator

We consider the following problems of the current simulator from sections 2.2 and 2.3:

Problem 1: Planning impassable actual paths An agent has a circular shape and a common size. However, these features are not considered when the simulator plans actual paths. The simulator may therefore plan an incorrect path surrounded by an area impassable for an agent such as that in Fig. 5.

Problem 2: Planning actual paths that have a dead end Coordinates that are most distant from blockages on the entrance/exit dividing line are selected as an entrance/exit point. The simulator may therefore plan an actual path that has a dead end despite the existence of a passable actual path as shown in Fig. 6.

Problem 3: Representation of traffic congestion An agent moves like a fluid in the simulation. In other words, agents move on a narrow road while pushing against each other as shown in Fig. 7. If agents consist of only humans, we expect that agents behave in this way. However, agents include robots, cars and other entities in reality. We therefore need confirm the availability of the current traffic simulator to various kinds of agents.

This paper focuses on problems 1 and 2. With respect to problem 3, we show a different approach in the Appendix, because it is difficult to simulate different types of agents at the same time.

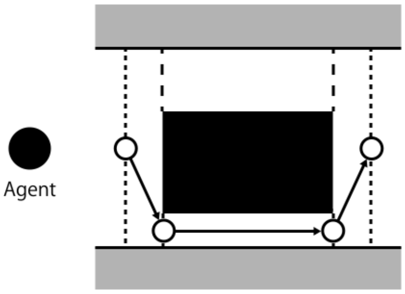

Fig. 5. Incorrect path 1

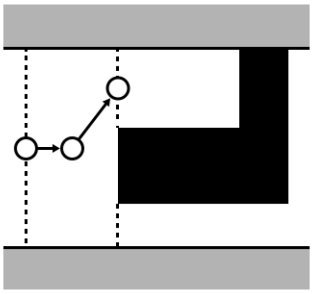

Fig. 6. Incorrect path 2

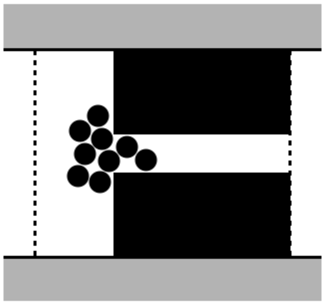

Fig. 7. Current traffic congestion

3 New Traffic Simulator

3.1 Purpose

We develop a new traffic simulator that improves the points mentioned in section 2.4. We thus aim to allow the automatic and appropriate movements of agents, and reduce the burden of competitors and researchers. The simulator maintains compatibility with the current simulator.

3.2 Approach

The current traffic simulator plans an impassable actual path as described in section 2.4. To solve this problem, the following conditions should be considered when the traffic simulator plans an actual path.

1. The shape and size of an agent
2. Selection of entrance/exit coordinates of roads based on the state of connection of roads

We can therefore solve the problem if the simulator derives actual paths considering the above points.

Shape and size of an agent Before planning an actual path, the simulator extracts a movable area for an agent according to the shape and size of the agent. This prevents the simulator from planning an impassable path; i.e., the simulator can derive actual paths by regarding an agent as a dot. For the road shown in Fig. 8, a movable area is shown as the shaded area in Fig. 9.

State of connection of roads To derive an actual path considering dead ends due to blockages as shown in Fig. 6, the simulator creates one graph by connecting graphs of the connecting parts of required roads. The simulator thus derives another actual path even if it encounters a dead end.

3.3 Control of agent movement

Our simulator uses the same flow of Fig. 1 in controlling agent movement. The member variables indicated in Table 1 also are used when agents move.

We describe ② in the figure in section 3.4. ① and ③ are the same as those of the current simulator.

3.4 Planning an actual path for each agent

Our simulator introduces the extraction of movable areas on roads and path planning after connecting graphs of required roads into the current simulator based the discussion in section 3.2.

Our simulator plans an actual path for each agent according to the following procedure.

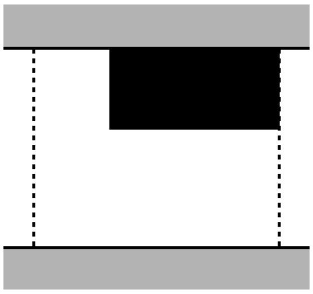

Fig. 8. Movable area before area extraction

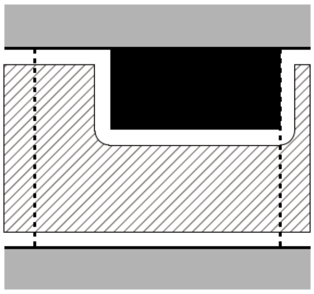

Fig. 9. Movable area after area extraction

1. The movable area of a road is extracted from each road specified by a sequence of roads.
2. Each movable area is decomposed into cells by drawing a line segment from each coordinate of its vertex along the Y axis (given by a geographical information system).
3. A graph is created. On the graph, the node is the center point of the line segment drawn in step 2. The edges of the graph are created by connecting these nodes.
4. The graphs created in step 3 are integrated into one graph
5. Path planning is performed on the graph and obtains the shortest actual path.

When a source or destination point of an agent is not on the graph, the point closest the source or destination on the graph is used as a relay point.

Furthermore, when the simulator extracts the movable area directly, curves appear as the boundary line of the area. In such a case, the simulator cannot apply the simple decomposing algorithm mentioned in section 2.2 and step 2 to the area. The simulator therefore uses the extracting method that does not generate curves as illustrated in Fig. 10. However, a passable actual path is sometimes regarded as an impassable path by the method. In this case, the simulator extracts an additional emphasized area as illustrated in Fig. 11, after normal area extractions.

3.5 Implementation

We will implement the traffic simulator using Java. The implemented simulator will be published in `https://github.com/miya224/TinyTraffic`.

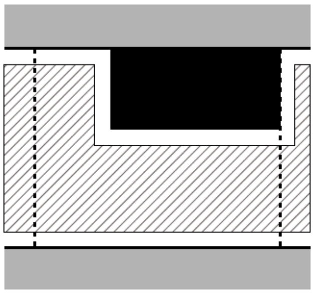

Fig. 10. Area extraction in our simulator

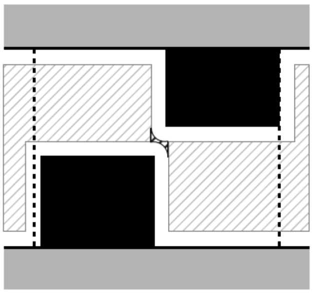

Fig. 11. Exceptional area extracting

4 Evaluation

4.1 Method

We have not yet completed the implementation of our simulator at the time of writing this paper. We therefore confirm expected operational examples of the simulator, and consider whether specifications satisfy requirements.

4.2 Operational examples of basic movements

We here describe operational examples of basic movements with only one agent.

Movement over a short range The simulator plans the actual path as shown in Fig. 12 when an agent wants to move over a short range as in the case of moving along the same road.

Movement over a long range The simulator plans the actual path as shown in Fig. 13 when an agent wants to move over a long range as in the case of moving along multiple roads.

4.3 Operational examples related to the problems of interest

We next describe operational examples of movements related to the problems of interest.

Movement on a road on which impassable paths also exist We consider an operational example in the environment as shown in Fig. 5 mentioned in the current traffic simulator. The new traffic simulator plans the actual path shown in Fig. 14 from the road.

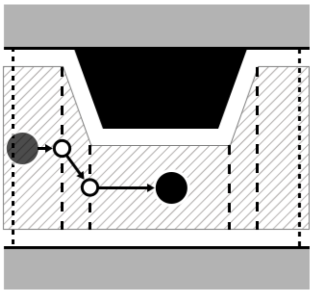

Fig. 12. Planning over a short range

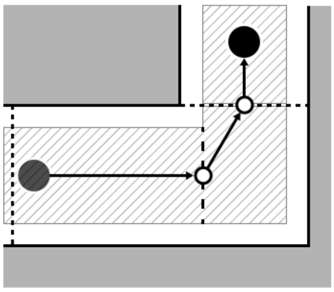

Fig. 13. Planning over a long range

Movement on a road along which agents may reach a dead end We consider an operational example in the environment as shown in Fig. 6. The new simulator creates the graph shown in Fig. 15. The simulator then performs path planning on the graph, and moves the agent along the actual path of the lower side.

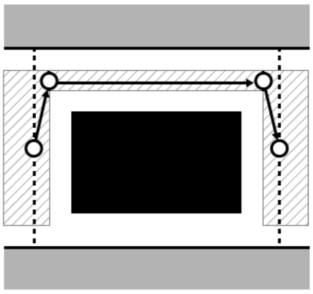

Fig. 14. Planning for problem 1

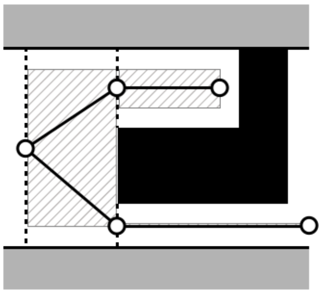

Fig. 15. Planning for problem 2

4.4 Operational examples of other special movements

We finally consider an example of a new problem that can occur for the new simulator.

Movement of multiple agents and multiple actual paths We consider an operational example in the case that multiple agents try to pass along a road for which there are multiple actual paths to the destination. Despite the fact that the agents can avoid traffic congestion if they select actual paths different from each other, they cannot do so in the new simulator because the simulator necessarily selects the shortest actual path.

4.5 Consideration

We know that our simulator can plan the actual paths that the current simulator can plan from examples of basic movements. Furthermore, we know that the new simulator also plans actual paths that the current simulator cannot plan from examples related to the problems of interest in the present study. Our simulator therefore satisfies all requirements.

However, our simulator faces a new problem in the case that multiple agents move along a road having multiple actual paths. The problem arises from the path planning algorithm used for planning actual paths. We can therefore solve the problem by introducing a path planning algorithm that considers detours.

5 Conclusion

We worked on improving the traffic simulator used in RoboCupRescue Simulation. We first listed problems of interest for the current traffic simulator. We next described that planning of actual paths different from that of the current simulator as a method of solving the problems. We finally described our new traffic simulator and its specifications introducing our method.

In an evaluation of our simulator, we confirmed whether the specifications allow the requested operations for various situations that occur in RoboCupRescue Simulation. We found that our simulator can perform the requested behaviors for all problems that we considered. In addition, a new problem arises because of the changed specifications, but we anticipate that we can solve it by changing the path planning algorithm.

We will finish implementing our traffic simulator according to the specifications by RoboCup IranOpen 2018 in future work and aim for the simulator to be adopted formally in RoboCupRescue Simulation.

Acknowledgement

We thank Okan Asik and Ali Modaresi for their advice in determining the specifications of our traffic simulator.

This work was supported by JSPS KAKENHI Grant Number JP16K00310 and JP17K00317, by the Nitto Foundation, and by Joint Usage and Research Center, Research Center for San-En-Nanshin Regional Collaboration, Aichi University.

Appendix

This section describes a traffic congestion model different from that of the current traffic simulator. This traffic congestion model is based on the graph model. The model regards an agent as a car, an edge of a graph as a lane, and a node of a graph as a junction, and reflects the limitations of movement in lanes/at junctions and the speed increase/decrease of cars depending on the lane size as follows.

Limitation for movement in lanes A car cannot overtake another car in the same lane. Therefore, an agent should not move faster than another agent located forward on the same edge traveling in the same direction.

Limitation for movement at junctions Cars pass through the same junction by mutual concession. If they try to pass through a junction at the same time, they will collide and cause an accident. Therefore, more than one agent cannot move on the same node at once.

The speed increase/decrease of cars depending lane sizes An actual car adjusts its speed not only at lanes or junctions but also by the visibility of its surrounding including the size of the road. Therefore, the speed of an agent is adjusted by the size of the area around its edge and the number of agents on the edge.

We next show an operational example of the model in the case that multiple agents try to pass along the same actual path at the same time. A traffic simulator that adopts this model transfers each agent as shown in Fig. 16→Fig. 17 according to the limitations of nodes/edges. In addition, the speed of each agent becomes low because multiple agents gather on the same edge.

We finally illustrate a simplified flow, which represents how to transfer agents along their actual paths using this model, in Fig. 18. Therefore, the speeds of agents are readjusted only when an agent reaches the next node/edge. We thus expect that the model can be implemented at a lower calculation cost than the current model.

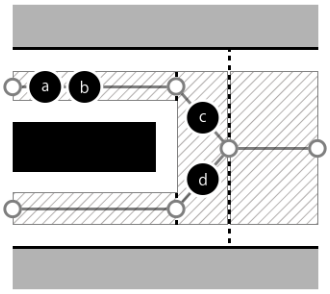

Fig. 16. Movement of multiple agent 1

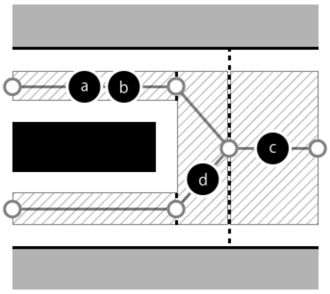

Fig. 17. Movement of multiple agent 2

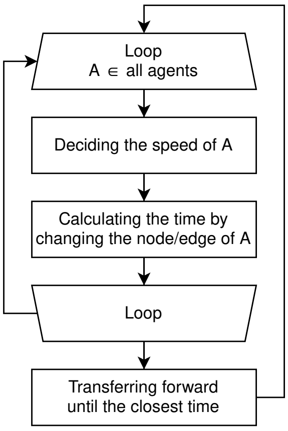
Loop
A ∈ all agents
Deciding the speed of A
Calculating the time by changing the node/edge of A
Loop
Transferring forward until the closest time

Fig. 18. Flow of a congestion model based on the graph model

References

1. RobocupRescue Simulation. `http://rescuesim.robocup.org/`
2. Okaya, M., Takahashi, T.: Representation of mental states at disasters and crowd evacuation simulation [in japanese]. Special Interest Group on Knowledge-Based Systems 88, 1–8 (2010)