RoboCup 2018 – TDP Rescue Agent Simulation AIT-Rescue (Japan)

Taishun Kusaka[1], Yuki Miyamoto[1], Akira Hasegawa[1], Shunki Takami[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. We designed and implemented a team based on the results of a combination experiment of various modules taken from teams that participated at RoboCup2017. We developed a new agent team with better modules in the Agent Development Framework. This paper presents the results of the combination experiment in detail. We confirm that our team based on the experimental results obtained a better score than the champion team at RoboCup2017.

1 Introduction

In the previous agent development environment of RoboCupRescue Simulation (RCRS)[2], it was difficult to understand codes of another team's agent program. To resolve this problem, the Agent Development Framework (ADF)[1] was introduced last year as an agent development environment. Readability agents became easier to understand owing to the modularity of the ADF. Furthermore, RRS-OACIS[3] was introduced as an experiment support application. RRS-OACIS allows a number of simulations to be run easily. In the present study, we first confirmed the effectiveness of various modules in many combination experiments and then developed a new team with good modules found in those experiments.

Chapter 2 describes the combination experiments and gives experimental results and considerations. Chapter 3 describes the features of each better module that we found in the experiment. Chapter 4 evaluates an agent developed with such found modules. The evaluation confirms that our team based on the research results scored better than the champion team at RoboCup2017.

2 Modules

2.1 Combination Experiments

We investigate how efficiently various modules used in RCRS contribute to rescue activities. We compare scores of all modules while exchanging a poper module of RoboCup2017 teams.

We ran all teams of RoboCup2017 in our experimental environment as a preliminary experiment. Target teams are the top five performing teams in the experiment: MRL, Aura, RoboAKUT, LarvicSaurus, and CSU_Yunlu. Because of the huge number of possible combinations, experiments are conducted for combination patterns of

1. BuildingDetector, RoadDetector, HumanDetector;
2. Search module for each agent;
3. ActionFireFighting, ActionExtClear, ActionTransport, ActionExtMove, and the corresponded PathPlanning modules.

Each module is evaluated for each team and each combination pattern to find the best combination of a team and modules. Only a VC3 map is used in RoboCup2017 because of the enormous number of combinations.

2.2 Results and Discussion of the Experiments

Tables 1, 2, and 3 give the results of the three teams having the highest experimental scores for the combination patterns described in 2.1. The top ranks in

Table 1: Top three combinations of pattern 1 and their scores

Rank	BaseAgent	BuildingDetector	RoadDetector	HumanDetector	Score
1	Aura	Aura	RoboAKUT	Aura	166.70
2	Aura	Aura	Aura	Aura	166.59
3	Aura	CSU_Yunlu	Aura	Aura	166.43

Table 2: Top three combinations of pattern 2 and their scores

Rank	BaseAgent	Search(FireBrigade)	Search(PoliceForce)	Search(AmbulanceTeam)	Score
1	Aura	sample	RoboAKUT	sample	179.31
2	Aura	sample	RoboAKUT	sample	178.30
3	Aura	CSU_Yunlu	MRL	sample	177.20

Table 3: Top three combinations of pattern 3 and their scores

Rank	BaseAgent	ActionFireFighting	ActionExtClear	ActionTransport	ActionExtMove	Score
1	Aura	RoboAKUT	RoboAKUT	Aura	Aura	175.69
2	Aura	LarvicSaurus	Aura	Aura	Aura	170.50
3	Aura	CSU_Yunlu	RoboAKUT	LarvicSaurus	Aura	164.61

the tables show that a combination of modules of various teams is effective. In other words, a well performing hybrid team can be developed with such modules to conduct effective rescue activities.

We therefore describe a hybrid agent with modules of Table 4 in later chapters.

Table 4: All adopted modules

Module Type	Team	PathPlanning's Team
BuildingDetector	Aura	-
RoadDetector	RoboAKUT	-
HumanDetector	Aura	-
Search(FireBrigade)	sample	-
Search(PoliceForce)	RoboAKUT	-
Search(AmbulanceTeam)	sample	-
ActionFireFighting	RoboAKUT	RoboAKUT
ActionExtClear	RoboAKUT	RoboAKUT
ActionTransport	Aura	Aura
ActionExtMove	Aura	Aura

3 Strategies

We designed and implemented our agent with modules found in the combination experiments according to the ADF. Our agent has features described according to source codes and the TDP[6][4] of corresponding teams in the following sections.

3.1 Fire Brigade

A Fire Brigade agent has four modules as shown in Table 4. Two Aura modules, BuildingDetector and ActionExtMove, are applied to the agent. The agent uses the Search module in the Sample agent and the module ActionFireFighting from RoboAKUT.

BuildingDetecter The module generates a group by updating information of the building received from the simulation kernel, and selects an extinguishing target from the group. All buildings and Fire Brigades are assigned to four groups for the task.

The burning buildings group is generated from

- buildings recognized by the agent recently,
- buildings that ignited recently, and
- buildings that changed temperature recently.

An example of a burning buildings group is shown in Fig. 1. We use an agent and burning building group for ease of explanation. All buildings are represented as having a square shape. The colors of buildings show the levels of fieriness.

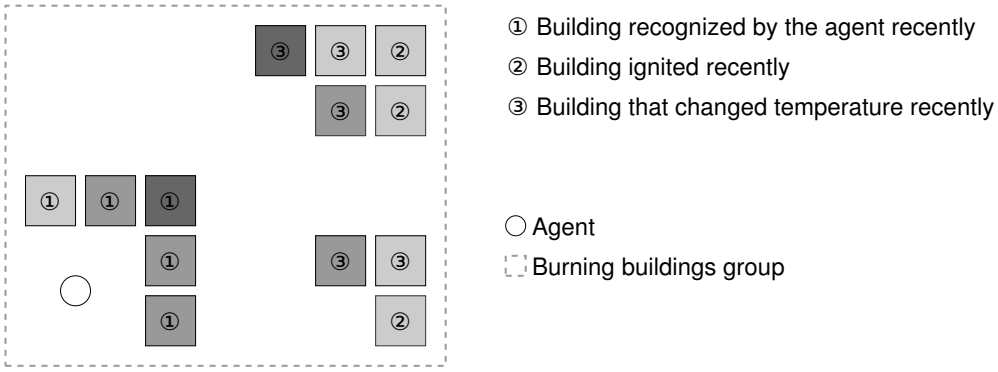

Fig. 1: Example of a fire group of BuildingDetector

An extinguishing target is the highest-priority building in the prioritized burning building group. Each building of the burning building group has a priority depending on the following degree and conditions.

1. Closeness of the boundary of the convex hull of the building, which is regarded as a corresponding vertex, and the burning building
2. Low degree of the building fieriness
3. Closeness of the building and its nearest gas station.
4. The number of burning buildings in the cell to which the building belongs, on a gridded map
5. Whether the group of the Fire Brigade and that of the building are the same.

Search When the agent cannot decide an extinguishing target, it needs to search for newer information. In such a case, the agent searches for information in the following two groups. Each agent is assigned a cluster by the clustering module.

1. All buildings of the agent's own cluster except for all refuges
2. All buildings except for all refuges

The search is first performed in group 1, and a searching target that is closest to the agent is selected by the path planning module. The search is performed in group 2 if the target is not selected in group 1.

ActionFireFighting The agent decides an action for an extinguishing target as shown in Fig. 2.

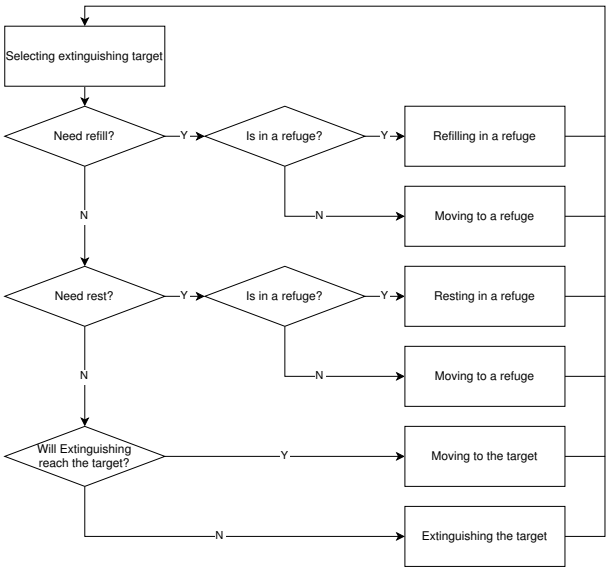

Fig. 2: Basic action decision of ActionFireFighting

ActionExtMove The agent calculates a path from its own location to the destination where the agent searches for information, and then moves to the destination. If there is no path, the agent calculates the path again without blocked roads and then moves along the path. The agent may search for information behind a blocked road if allowed.

PathPlanning In ActionFireFighting, the shortest path is calculated using the Dijkstra algorithm[5]. Moreover, the priorities depending on the fieriness of buildings and recognized information of the agent are provided to the Dijkstra algorithm in ActionExtMove.

3.2 Police Force

Table 4 shows that a Police Force agent has four modules. The three RoboAKUT modules, RoadDetector, Search and ActionExtClear are applied to the agent. Only the module ActionExtMove is from Aura.

RoadDetector The module selects a road in a cluster as a clearing target according to the following priorities ranked in descending order. The Police Force has already been assigned by a clustering module.

1. The road closest to the agent that receives a clear command
2. A road closer to an agent, where other agents are surrounded by blockades

3. A road in the cluster that satisfies one of three conditions: - a road near the refill point or destination selected by a Fire Brigade or an Ambulance Team - a road near the refuge that is the closest to the initial position of the agent - a road on the path to a hydrant that a Fire Brigade has selected for refilling tanks
4. The closest road that the agent has already passed, selected randomly when there is more than one closest road

If there are plural appropriate roads, the path planning module selects one road from them.

Search The module performs in the same way as the Search module in Fire Brigade.

ActionExtClear When the module receives the clearing target, the agent tries to remove blockades between it and the target. In removing blockades along the path to the clear target, the agent mainly removes the blockades that are on a line connecting the center points of the intersections of adjacent roads. If the blockades are within the clearing range of the agent, the agent removes them promptly; otherwise the agent moves to a point near them. If there are other agents near the agent, surrounded by a blockade, the agent preferentially removes the blockages. If some other agents are surrounded by a blockade, the agent prioritizes the target nearby. If the agent is likely exhausted, it moves to a refuge to recover.

ActionExtMove The agent moves to a position near the search target according to the module. If the module cannot find a path to the position, it sets a new reachable destination near the position. If the agent likely has no strength, it moves to a refuge to recuperate.

PathPlanning Modules are used in ActionExtClear and ActionExtMove. The module in ActionExtClear creates a path in the same way as the module in ActionFireFight. The module in ActionExtMove finds a path like the module in ActionExtMove of Fire Brigade.

3.3 Ambulance Team

Table 4 shows that an Ambulance Team agent has four modules. The three Aura modules—HumanDetector, ActionTransport, and ActionExtMove—are applied to the agent. The agent uses the Search module in the Sample agent.

HumanDetector The module choses an agent as a rescue target according to the following priorities ranked in descending order.

1. A Fire Brigade, Police Force or Ambulance Team is located within a certain range and buried but can be rescued before its health points are exhausted.
2. A Civilian, whose degree of burial is low, is within a certain range from the agent and satisfies one of four conditions:
   - it is located in a building and injured but not buried;
   - its strength is above a certain level and it can be rescued before becoming exhausted; it is buried but not located in a burning building.
3. A Civilian does not satisfy only the degree of burial under priority condition 2 above.
4. A Civilian on a road is not buried but is exhausted before the termination of the simulation.

If there are plural target agents with the same priorities, the path planning module selects one of them.

Search The module performs in the same way as the Search module in Fire Brigade.

ActionTransport The module decides which action the agent should perform to rescue its target. The decision making processes are shown in Fig. 3. However, if the relief target satisfies all of the following conditions, it is temporarily left on a road without being transported to a refuge.

- Its health points are not exhausted before the termination of the simulation.
- The distance from the current position of the agent to the nearest refuge is more than a certain criterion.
- The current position is far enough away from a gas station that might explode within a certain time.

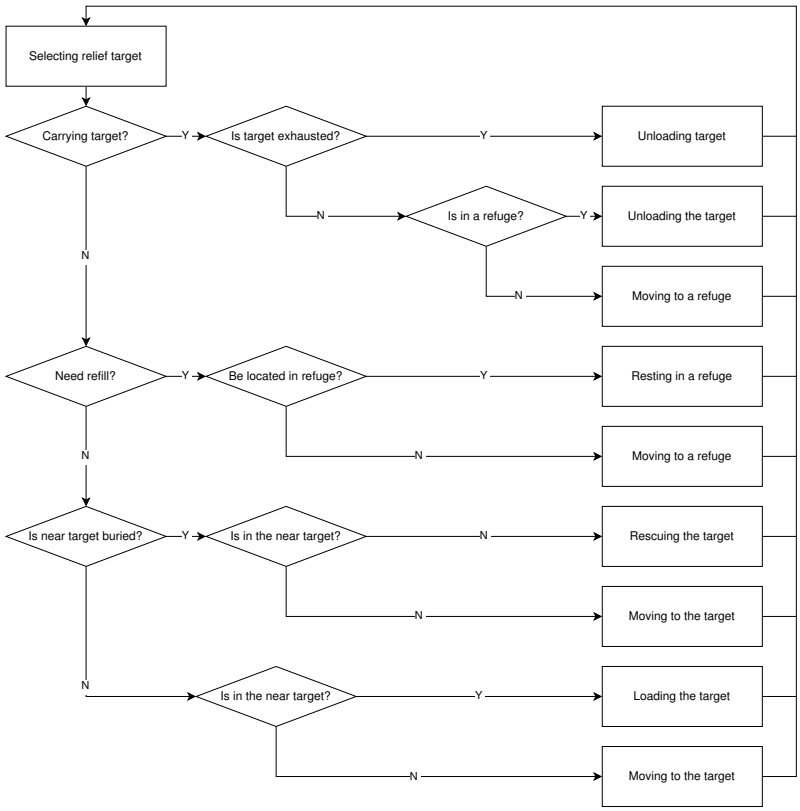

Fig. 3: Basic action decision of ActionTransport

ActionExtMove The module performs the same way as the module ActionExtMove of Police Force.

PathPlanning The module is used in ActionTransport and ActionExtMove. In both modules, it finds a path the same as the module ActionExtMove of Fire Brigade.

4 Preliminary Results

4.1 Experimental aim and content

The aim of the experiments is to confirm that agents with the combined modules proposed in Section 3 can carry out rescue efforts efficiently. The experiments compare the scores of our team with those of a sample team and the top five teams at the RoboCup 2017 competition.

4.2 Results

Results of the experiments are given in Table 5.

Table 5: Result of simulation

Team	**Map**				
	Elndhoven3	**Istanbul2**	**Paris2**	**SF3**	**VC3**
Our Team	83.3	17.2	40.6	42.3	132.3
Aura	115.7	32.9	48.3	46.9	170.3
CSU_Yunlu	95.5	11.6	15.2	52.5	33.6
LarvicSaurus	93.8	15.1	15.1	49.5	31.7
MRL	82.2	8.3	43.7	56.8	8.6
RoboAKUT	93.8	12.7	37.5	44.4	142.9
Sample	83.6	13.9	16.9	48.4	50.8

4.3 Considerations

Table 1 shows that the scores of our team are higher than the scores of the team that won the RoboCup 2017 competition (MRL) except in SF3. Our team also surpasses other teams on some maps. The results suggest that we can generate better agents by combining agents from different teams.

Furthermore, the results show an overall trend that each team has affinity with the maps. The trend means that there is an effective strategy for each map. It is thus necessary to perform simulations on many maps with many teams to generate better agent teams.

We will improve our teams by finding better combinations in further experiments.

5 Conclusions

We compared various rescue strategies in module units by simulating the replacement of specific types of modules among agents and generated a new agent from the results. Experiments show that the new combined agents can carry out rescue efforts efficiently.

It is clarified that there is an effective rescue strategy for each map. Therefore, to generate a better agent, the combination of modules on various maps needs to be reviewed.

References

1. Agent Development Framework. `https://github.com/RCRS-ADF/RCRS-ADF/releases`
2. RobocupRescue Simulation. `http://rescuesim.robocup.org/`
3. RRS-OACIS. `https://hub.docker.com/r/rrsoacis/rrsoacis/`

4. Akin, H.L., Asik, O.: RoboCup 2017 Rescue Simulation League Team Description RoboAKUT (Turkey) (2017)
5. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik 1(1), 269–271 (Dec 1959)
6. Ghahramanpour, M., Absalan, A., Kandeh, A.: RoboCup 2017 Rescue Simulation League Team Description Aura (Iran) (2017)