RoboCupRescue 2025
TDP Agent Simulation AIT-Rescue (Japan)

Ryosuke Suzuki[1], Yuki Shimada[1], Shuntaro Fujii[1], Ryuki Yamamoto[1], Ami Kobayashi[1], Kazutaka Kenmochi[1], Manaya Tsuzuki[1], Shintaro Ito[1], Syunya Niimi[1], Tomoki Mizuno[1], Yuto Shibata[1], Yuta Haba[1], Yuichiro Hoshi[1], Keisuke Ando[1], Chiharu Tomioka[1], Takeshi Uchitane[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

Abstract. One area of recent AIT-Rescue research is task assignment using the Response Threshold Method. Specifically, we applied the Response Threshold Method to the task assignment performed by agents. We then verified its effectiveness in communication-restricted environments. Experimental results confirmed that the Response Threshold Method is effective in some scenarios where agents are concentrated. However, in other scenarios, it underperformed the Greedy Algorithm. Additionally, for this year's agent competition, we introduced a new strategy for the police force. Specifically, this strategy involves performing debris cleaning along priority roads frequently used for transport. This allows transport routes from various locations to be secured early. Also, for the implementation of this strategy, we implement a module that forms priority roads frequently used for transport.

1 Introduction

AIT-Rescue is a team that has participated in the agent competition of the RoboCup Rescue Simulation League since its first edition. Last year's competition employed a strategy of distributing agents throughout the city to promote rescue activities in various locations. As a result, we won the first prize in last year's competition. In addition to these efforts, AIT-Rescue is also conducting research on Multi-agent systems using RoboCup Rescue Simulation (hereafter RRS).

Recent AIT-Rescue research includes a study of task allocations using the Response Threshold Method. RRS may assume a disaster environment where communication among agents is not possible.

The Greedy Algorithm has traditionally been used for task allocation in such environments. On the other hand, other task allocation methods that can operate in similar environments include the Response Threshold Method. However, no studies have been identified in which the Response Threshold Method was applied to task allocation in environments where inter-agent communication is not possible. However, no studies have been identified in which the Response

Threshold Method was applied to task allocation in environments where inter-agent communication is not possible. Therefore, we realized an agent that allocates tasks according to response threshold in this study. The effectiveness of the Greedy Algorithm was verified by comparing it with the Greedy Algorithm in an environment where communication was impossible. As a result, we confirmed a tendency to outperform the Greedy Algorithm agents in some scenarios where agents are concentrated.

In preparation for the competition, the Police Forces also are developing a new strategy. Specifically, the strategy is to carry out debris cleaning along priority roads that are frequently used for transportation. In addition, a module is implemented to form priority roads that are frequently used in transport to realize this strategy.

Chapter 2 describes the design of agents employing the Response Threshold Method and experiments comparing rescue outcomes with Greedy Algorithm agents in environments where communication is not possible. Chapter 3 describes the problems with last year's Police Forces and the modules to be incorporated in the new Police Forces strategy. Chapter 4 describes each agent's strategy for this year's competition. Section 5 summarizes this TDP and describes the issues to be addressed until this year's competition.

2 Scientific challenge

2.1 Background

In the RRS, there are scenarios that assume disaster environments where communication between agents is unavailable [2]. In such communication-restricted environments, agents are unable to exchange information with one another. Therefore, in the competition, a Greedy Algorithm has traditionally been used as a task allocation strategy that can function under these conditions. In this Algorithm, tasks are executed in order of their priority, starting from the highest. Here, a task refers to actions such as the rescue or transport of civilians by disaster relief agents.

In addition to the Greedy Algorithm, there is another task allocation strategy called Response Threshold Method, which can also work in environments without agent communication. The Method is inspired by the division of labor observed in social insects [1]. In this Method, an agent decides whether to perform a task based on two values: the stimulus, which represents the priority of the task, and the response threshold, which indicates how sensitively the agent reacts to the stimulus.

There have been no known applications of the Response Threshold Method in such communication-restricted in RRS. Therefore, in this study, we first develop agents that implement the method. Then, we compare the performance of these agents with those using the Greedy Algorithm under such environment. Through this comparison, we aim to evaluate the effectiveness of this method.

2.2 Response Threshold Method

Basic Principle The Response Threshold Method is based on two important concepts: stimulus (s) and response threshold (θ). Stimulus (s) represents the urgency or priority of a task. The response threshold (θ) reflects the agent's responsiveness to the stimulus. The probability $T_\theta(s)$ that an agent selects a task is determined by the following response function:

$$T_\theta(s) = \frac{s^n}{s^n + \theta^n} \tag{1}$$

Here, n is a parameter that controls the steepness of the probability change in response to the stimulus (In this study, the parameter n is set to 2). As shown in Equation (1), if the stimulus s is significantly smaller than the threshold θ, $T_\theta(s)$ approaches 0. This indicates a low probability of task selection. Conversely, if s exceeds θ, $T_\theta(s)$ approaches 1. This implies a high probability of task selection.

Probabilistic Task Selection The distinctive feature of the Response Threshold Method is that task selection is probabilistic. Agents may probabilistically ignore perceived tasks or may probabilistically select tasks with lower priority. This inherently stochastic nature distinguishes the method from deterministic approaches such as Greedy Algorithm.

2.3 Research Requirements

In this study, the agents consist only of fire brigades and ambulance teams, and the environment is set up without any debris. The environment excludes debris, as it is difficult to define debris removal as a standalone task relative to rescuing or transporting civilians. Additionally, communication between agents is disabled in order to evaluate the effectiveness of the Response Threshold Method under communication-restricted conditions. In addition, the definitions of the stimulus and response threshold are described below.

Definition of Stimulus The stimulus(s) is defined as a binomial function calculated from the remaining life span of a victim and the task execution time. It is designed to reflect the urgency of a task. Depending on the task type, two different stimulus functions are defined as follows:

- **Stimulus for rescue tasks (s_{fb}):**

$$s_{fb} = w_1 \cdot a + w_2 \cdot b \tag{2}$$

  where a is the remaining life span of the victim, b is the time required for rescue, and w_1 and w_2 are weights.
- **Stimulus for transport tasks (s_{at}):**

$$s_{at} = w_3 \cdot a + w_4 \cdot c \tag{3}$$

  where a is the remaining life span of the victim, c is the time required for transport, and w_3 and w_4 are weights.

The weights (w_1, w_2, w_3, w_4) and the response threshold are treated as tunable parameters, which are experimentally adjusted to achieve optimal performance under various disaster scenarios.

Definition of Response Threshold The response threshold (θ) is set to a fixed value for each type of agent (fire brigade and ambulance team). This simplification is based on the assumption that agents of the same type possess equivalent capabilities in the RRS environment.

2.4 Agent Design Based on the Response Threshold Method

The agent architecture based on the Response Threshold Method is designed with the following functional modules:

1. **Information Exploration Module**: Agents patrol buildings to discover tasks, following routes optimized using the 2-opt algorithm. This module is responsible for task discovery in the environment.
2. **Task Selection Module**: Once a task is discovered, the agent uses the Response Threshold Method to probabilistically decide whether to engage in the task. This module implements the response function defined using the stimulus functions in Equations (2) and (3), and the response function in Equation (1).

Determining Response Threshold Parameters This study treats the response threshold θ as a fixed value. The determined response thresholds are shown in Table 1. Specifically, in preliminary experiments, we tested several candidate values (1, 5, 10, 50, 100) in rescue simulations. Among them, the value $\theta = 5$ achieved the highest rescue score and was adopted as the fixed threshold for both fire brigades and ambulance teams in the main experiments. Although the response threshold was set separately for each agent type, the same value was applied to all agents of the same type, based on the assumption that agents of the same type are assumed to have identical capabilities within the RRS environment.

A fixed threshold was used to verify the fundamental effectiveness of the Response Threshold Method. However, since the optimal value of the response threshold may vary depending on environmental factors such as the disaster scenario or the number of agents, introducing a dynamic adjustment mechanism for the threshold remains an important direction for future work.

Determining Weight Parameters The weights (w_1 to w_4) in the stimulus functions are parameters that adjust the relative importance of each term (such as the victim's life span or the task execution time). The determined weights are shown in Table 1. In this study, these weights were also treated as fixed values. The values were determined using Bayesian optimization. Specifically, Bayesian optimization was performed in a disaster scenario with 300 tasks, aiming to maximize the rescue score. The resulting optimal weight values were used as fixed values in the main experiments.

The reason for using Bayesian optimization is that the parameter space for the weights is large, making exhaustive search impractical. Bayesian optimization was chosen as an efficient search method. Similar to the response threshold, the weights were fixed to prioritize verifying the basic effectiveness of the Response Threshold Method under static conditions. However, since optimal weight values may vary depending on environmental factors such as the disaster scenario or the number of agents, incorporating a dynamic weight adjustment mechanism is also a future challenge.

Table 1: Agent parameters used in the experiment

Parameter	Value
Response threshold of fire brigades	5
Response threshold of ambulance teams	5
Weight w_1	-0.973
Weight w_2	0.884
Weight w_3	0.127
Weight w_4	-0.229

2.5 Purpose of the Experiment

The purpose of this experiment is to evaluate the effectiveness of the Response Threshold Method for task allocation in an RRS environment where communication between agents is unavailable. To achieve this, we compare agents using the Response Threshold Method with those using the Greedy Algorithm in various disaster scenarios.

2.6 Experimental Conditions

The researchers used map data from the Sakae district of Nagoya city. Eight refuges were distributed throughout the city. The number of available beds in each refuge was set to 300, which is the maximum number of civilians. The researchers deployed 75 fire brigade agents and 25 ambulance team agents as disaster relief agents. The movement speed for all agents was 40 m per timestep. Each agent's perception range was a circular area with a radius of 30 m. The simulation execution time was set to 300 timesteps. Additionally, the researchers set the following two conditions as variable factors.

Degree of Distribution for Agent and Civilian Placement The degree of distribution is a key factor for adjusting the difficulty of rescue activities. We varied the degree of distribution in the following three levels.

- **Centralized Placement**: Agents and civilians are concentrated in a specific narrow area.

  - **Intermediate Placement**: Agents and civilians are placed in an intermediate area of the city.
  - **Distributed Placement**: Agents and civilians are distributed throughout the entire city.

Number of Civilians The total number of civilians in the city was varied across five levels: 100, 150, 200, 250, and 300.

By combining these variable factors, a total of 45 disaster scenarios were created (3 agent placement patterns × 3 civilian placement patterns × 5 levels of civilian numbers). These experimental settings allow the researchers to analyze how the probabilistic task-ignoring characteristic of the Response Threshold Method is affected by changes in the number of tasks (i.e., the number of civilians). Furthermore, these settings enable a comparison between the two methods across various disaster scenario patterns.

Comparison Agents We implemented two types of agents.

- **Response Threshold Method Agent**: This agent adopts the Response Threshold Method and is configured with the parameters in Table 1.
- **Greedy Algorithm Agent**: An agent where only the task selection module was changed to use a Greedy Algorithm. This agent quantifies tasks by squaring their stimulus values and selects the task with the highest resulting value. This ensures a direct comparison with the stimulus used in the Response Threshold Method.

Both agent types share the same architecture, except for the task allocation mechanism. This enables a comparison focused on the impact of the task allocation strategy.

2.7 Experimental Results

Overall Trend Based on the experimental results in Table 2, the researchers performed a Wilcoxon signed-rank test. The results showed that the rescue score of the Greedy Algorithm Agent was statistically significantly higher than that of the Response Threshold Method Agent overall. This result suggests that the Greedy Algorithm is a more effective task allocation method for the overall disaster scenarios set in this experiment.

Results per Scenario In specific scenarios, particularly those with "Centralized Agent Placement and Centralized Civilian Placement," the researchers observed cases where the Response Threshold Method Agent achieved a higher rescue score than the Greedy Algorithm Agent. Such scenarios accounted for 10 out of the 45 total scenarios. Furthermore, in many other scenarios, especially those involving "Distributed Agent Placement" or "Centralized Civilian Placement," the tendency for the Greedy Algorithm Agent to show a greater rescue score than the Response Threshold Method Agent was prominent. These scenarios constituted the majority, accounting for 35 out of the 45 scenarios.

Table 2: Rescue score of the Response Threshold Method Agent relative to the Greedy Algorithm Agent for each disaster scenario

Placement Agent	Civilian	100	150	200	250	300
Centralized	Centralized	0.942	1.114	0.975	0.880	0.981
	Intermediate	1.016	0.988	0.915	1.129	0.906
	Distributed	1.065	1.020	0.863	0.889	1.011
Intermediate	Centralized	0.956	0.946	0.922	1.037	0.925
	Intermediate	1.014	0.938	0.912	0.861	0.937
	Distributed	0.977	0.964	1.009	0.937	0.931
Distributed	Centralized	0.844	0.808	0.933	0.867	0.852
	Intermediate	0.877	0.994	0.911	0.906	0.915
	Distributed	0.860	0.855	1.016	0.946	0.969

2.8 consideration

Scenarios Where the Response Threshold Method Was Effective and the Reason The Response Threshold Method agent tended to perform better than the Greedy Algorithm agent in scenarios with centralized placement of both agents and civilians. This is because the Response Threshold Method agent has the property of probabilistically ignoring tasks, which made it possible to prevent agents from focusing excessively on tasks near their initial positions. As a result, they were likely able to rescue civilians distributed over a wide area. In contrast, Greedy Algorithm agents always choose the highest-priority task, which tends to make them focus on tasks near the starting point. This may have delayed responses to distributed civilians.

Scenarios Where the Greedy Algorithm Was Effective and the Reason The Greedy Algorithm agent tended to outperform the Response Threshold Method agent in scenarios with distributed placement of agents or centralized placement of civilians. This is because the Greedy Algorithm agent reliably executes tasks with the highest priority, allowing for efficient use of limited rescue resources. In particular, when agents are distributed or tasks are concentrated in specific areas, it is important to quickly handle high-priority tasks. The Greedy Algorithm effectively met this requirement. On the other hand, the Response Threshold Method agent may delay processing high-priority tasks because it probabilistically ignores some tasks. As a result, differences in rescue outcomes were observed.

Characteristics and Challenges of the Response Threshold Method The probabilistic task-ignoring behavior, which is a feature of the Response

Threshold Method, has both advantages and disadvantages. It helps prevent agents from over-concentrating on tasks. However, it may also delay the processing of high-priority tasks.

It was also revealed that the effectiveness of this method depends heavily on the disaster scenario. Although it may work effectively under specific conditions, such as centralized placement of both agents and civilians, it often performs worse than the Greedy Algorithm in general environments.

3 Modules

3.1 Problems of the Police Force Corps in RoboCup2024

The AIT-Rescue Police Force in RoboCup 2024 employs a distributed debris cleaning strategy for each clustered area. The advantage of this approach, achieved by partitioning the area, is that it allows agents to operate concurrently in different locations, thereby increasing overall clearing efficiency.

Although the team won the competition, their debris cleaning activities did not contribute to rescuing civilians. This stemmed from the inability of the Police Force to secure passable routes connecting buildings containing injured civilians to designated refuges.

Key characteristics of the scenarios used in last year's competition included the distributed deployment of civilians and agents, and a large amount of debris. This issue was particularly observed on scenarios with significant amounts of debris. As an example, the kobe2 scenario used in last year's competition is shown in Figure 1. Furthermore, Figure 2 shows the situation on the kobe2 scenario around Step 160, where this problem was observed.

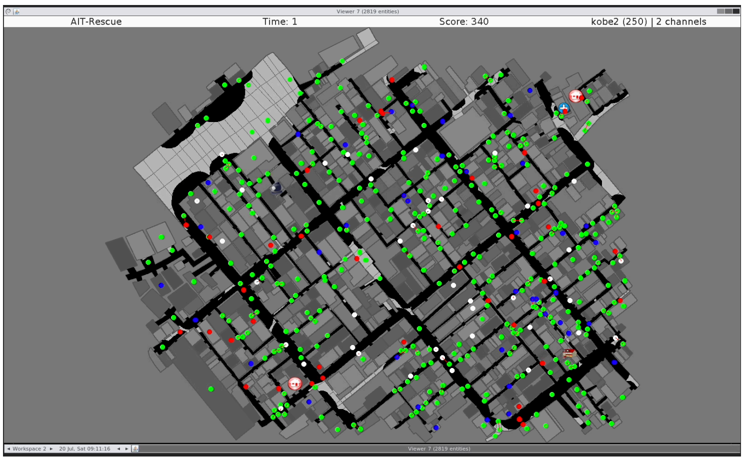

Fig. 1: the kobe2 scenario

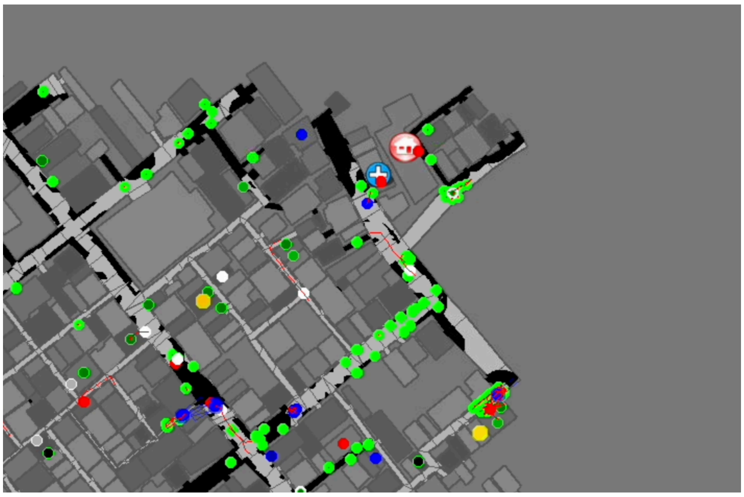

Fig. 2: around 160 Steps in kobe2

In scenarios with a large amount of debris, such as the one shown in Figure 1, it is difficult for the Police Force to clear all debris before the simulation ends. Also, Police Force agents that start buried at the beginning of the simulation become incapacitated, meaning that in some cases, parts of the necessary transport routes may remain uncleared. Furthermore, since the Police Force tends to clear nearby debris without specific route prioritization, even if they complete clearing their assigned area, they do not necessarily move towards areas containing parts of the required transport routes. Consequently, the approach of simply dividing the map into multiple areas and performing decentralized clearing can lead to situations where necessary transport routes are not secured. Figure 2 shows an example of such a situation, where the area around the refuge remains uncleared despite being near the end of the simulation.

This problem prevented the Ambulance Team from transporting civilians to the refuge, resulting in debris cleaning activities that ultimately did not contribute to civilian rescue. Therefore, the Police Force needs to clear roads not just by working decentralizedly within assigned areas, but by specifically considering and prioritizing the routes from buildings containing civilians to the refuge.

3.2 Lifeline Graph Module

As described in section 3.1, in a scenario with a large amount of debris, the current Police Force cannot provide adequate transport routes from buildings to the refuge. This is because the Police Force is focused on the debris cleaning of the entire area. In a scenario with a lot of debris, it would take a lot of time for debris cleaning in the area in charge, and some Police Forces would be inoperable due to debris. Therefore, debris cleaning is not possible in all areas, and in some cases, parts of the transport routes may remain unrevealed.

Furthermore, since the Police Force engages in debris cleaning haphazardly, even after completing debris cleaning in their assigned areas, they may not proceed to areas that include sections of the transport routes.

To secure transport routes, a strategy of debris cleaning along priority roads is introduced to the Police Force this year. As a concrete strategy, priority roads are determined in advance, and each Police Force conducts distributed debris cleaning along them. Priority roads refer to roads that are frequently used for transport routes. Compared to conventional strategies, transport routes can be secured more efficiently and earlier by having the Police Force advance debris cleaning along priority roads.

This section describes the Lifeline Graph Module to be implemented to realize the strategy. The Lifeline Graph Module generates prioritized roads for transport routes.

Priority roads to be formed The Lifeline Graph Module defines priority roads based on Major Intersections and Lifeline Roads. Table 3 shows the definitions of these terms.

Table 3: Names and Definitions of major Areas

Name	Definition
Major Intersections	Intersections are expected to be frequently used for transport to refuges
Lifeline Roads	Roads that must be secured to transport civilians to refuges. Established as routes among Major Intersections and refuges, or among Major Intersections

The prioritized roads must provide access from any area on them to all refuges. This is because, in rescue activities, civilians are transported to different refuges depending on the situation. Therefore, the Lifeline Graph Module forms priority roads that ensure access to all refuges via Major Intersections.

Procedure for forming priority roads In the Lifeline Graph Module, priority roads are formed from a group of roads by the following procedure.

1. Calculate Major Intersections
2. Connections among major areas

In Step 1, Major Intersections are calculated. In the calculation of Major Intersections, we first calculate the shortest routes from areas where the transport of citizens is likely to occur to the refuge. The route calculated here is the route from the Ambulance Team, the Fire Brigade, and the building to the refuge. However, because calculating all the routes from a building to the refuge would be computationally expensive, buildings are randomly selected to limit the number of routes. In addition, the reason for considering a route starting from the

Ambulance Team and the Fire Brigade is that a civil rescue is expected to occur in the vicinity of the route.

Next, the number of intersecting roads and the number of intersections in the calculated multiple routes are determined. Then, roads with several intersections greater than or equal to the threshold are designated as Major Intersections. This threshold is given as a hyperparameter.

In Step 2, priority roads are formed by connecting major areas. In connecting major areas, major areas are connected in the order of the shortest path and the shortest route distance. Therefore, the route distances from each point in the formed priority road network to the refuge are kept relatively short.

The connection procedure among major areas first selects the pair with the shortest path distance from the refuge and the group of Major Intersections. Next, the selected pair and the path among the two points are connected and graphed. The path among the two points is the Lifeline Road. The same pair selection and connection procedure is then repeated with the formed road graph as a pair selection option. Finally, the formation of a priority road is completed when it becomes a single road graph.

4 Strategies

4.1 Police Force

The Police Force is responsible for clearing debris scattered on the roads and making the roads passable. By making the roads passable, other agents can reach their destinations, leading to the discovery and rescue of civilians. However, in a city with a lot of debris, it is difficult to remove all the debris during simulation time. Therefore, the number of roads that can be made passable by the end of the simulation is limited. Accordingly, the Police Force needs to select effective roads that will lead to the rescue of civilians and clear debris on a priority basis.

As described in section 3.1, the current strategy of the Police Force does not allow for the securing of sufficient transport routes in environments with a large amount of debris. At last year's competition, this issue delayed the transport of civilians to the refuge. Therefore, this year's AIT-Rescue Police Force will introduce a strategy focused on securing transport routes.

In this strategy, priority roads are first determined before debris cleaning begins. Then, each Police Force will carry out debris cleaning along these priority roads in a decentralized manner. The priority roads are formed as roads that are frequently used for transportation to the refuge. Therefore, clearing debris along the priority roads leads to sucuring of rapid and effective transport routes. In addition, the Police Force carries out debris cleaning in a way that extends the priority roads. This ensures transport routes to the refuge can be secured from various locations. The specific process of debris cleaning by the Police Force in this strategy is as follows.

1. Formation of priority roads by the Lifeline Graph Module
2. Assignment of Police Force to priority roads

3. Debris cleaning centered on priority roads by each Police Force

In Step 1, priority roads are formed by the Lifeline Graph Module. The Lifeline Graph Module generates priority roads based on intersections that are expected to be frequently used for transportation to refuges. As a result, the generated priority roads form a road structure that enables quick and reliable access from various locations to multiple refuges.

In Step 2, Police Force is assigned to the generated priority roads. Here, the roads along the priority roads are defined as Lifeline Roads. In this strategy, the Police Force conducts debris cleaning along these Lifeline Roads. Therefore, each Police Force must first reach a lifeline road. Accordingly, in Step 2, each Police Force is assigned to the nearest lifeline road to determine the starting point for debris cleaning.

In Step 3, each Police Force clears debris on the priority roads, starting from the assigned lifeline road. Through this process, the Police Force aims to secure transportation routes quickly and effectively. However, clearing debris only the priority roads. is not sufficient to fully resolve situations in which Ambulance Team and Fire Brigade are immobilized due to debris. Additionally, securing access routes into buildings remains a challenge. Therefore, the Police Force must also address these issues while continuing to clear debris on the priority roads. In this strategy, when multiple Police Force converge on a priority road, one Police Force continues clearing debris on the priority road, while the other engages in supplementary debris cleaning such as resolving stuck emergency vehicles and securing access to buildings. This enables the parallel realization of both transportation route clearance and support for individual rescue operations.

4.2 Ambulance teams

Ambulance teams are responsible for transporting injured civilians to the most suitable refuge. The strategy of the ambulance teams requires careful selection of non-overlapping civilians and the choice of refuges considering the availability of beds. This is crucial because if multiple ambulance teams target the same civilian, the rescue of other transportable civilians may be delayed. Furthermore, if a refuge is full, the physical condition of the civilians waiting for an available bed may deteriorate, potentially leading to their death while waiting.

Therefore, the ambulance team of AIT-Rescue adopts a strategy that focuses on selecting civilians to transport and predicting the conditions of refuges. Specifically, the team realizes efficient civilian transport by selecting transport targets based on perceived information at each step and shared information via communication, and by predicting the state of refuges using a queueing model. In the following, we describe how the refuges are selected.

Refuges are selected using a Greedy Algorithm, where the evaluation value is defined as the time until civilians can use an available bed. With this method, the refuge that minimizes the time for a civilian to enter is predicted, and the best choice is made at each step. The time until a civilian can use an available bed is determined by the longer of the travel time to the refuge and the expected waiting time after arrival. This is because the waiting time at the shelter decreases with the time required for transport.

The travel time to a refuge is calculated by dividing the path distance to the refuge by the movement speed of the ambulance team. The path distance is based on the route computed by the path planning module. Specifically, the average movement speed is obtained by repeatedly moving an ambulance team between random locations. This approach allows us to reflect various road conditions on average and obtain travel times that do not depend on specific road situations.

The waiting time at refuges is estimated using the $M/M/m$ queuing model. This model represents a situation where m service counters process a single queue in parallel. In this strategy, the model is applied to a situation in which transported citizens are placed in a waiting state in several hospital beds.

The average waiting time W at a refuge is calculated by the following equation (4):

$$W = \pi_0 \frac{\rho(c\rho)^c}{\lambda(1-\rho)^2 c!} \tag{4}$$

Furthermore, the probability π_0 that there are zero waiting civilians is calculated by the following equation (5):

$$\pi_0 = \left[\left(\sum_{k=0}^{c-1} \frac{(c\rho)^k}{k!} \right) + \frac{(c\rho)^c}{c!} \frac{1}{1-\rho} \right]^{-1} \tag{5}$$

Here, c is the number of beds in a given refuge, λ is the number of civilians transported to that refuge per unit time, μ is the average number of civilians discharged from beds at that refuge per unit time, and ρ is the utilization rate calculated as λ/μ, representing how many civilians are transported to the refuge while one civilian finishes treatment. Notably, the value of λ is obtained from preliminary experiments. For μ, we use the inverse of the average damage of the civilians currently at the target refuge, based on past observations.

From the above, efficient transportation of civilians is achieved by predicting the waiting time at refuges using Equation (4).

4.3 Fire Brigade

The Fire Brigade is responsible for rescuing agents buried in collapsed buildings. A key strategy for the Fire Brigade is prioritizing rescue of other rescue agents and designing a robust rescue strategy adaptable to the changing disaster situation. This is crucial because the disaster situation evolves dynamically. Unplanned searches risk missing rescue opportunities. Furthermore, the RRS requires rescuing as many civilians as possible before the simulation ends. Therefore, when the number of active rescue agents is low, rescue efficiency decreases, potentially reducing the total number of rescued civilians.

Thus, the AIT-Rescue Fire Brigade dynamically determines which buildings to explore and prioritizes rescue targets. Specifically, it uses information such as civilian rescue requests and building exploration status for dynamic target selection. Prioritizing the rescue of other rescue agents allows for efficient building exploration and civilian rescue. The prioritization of rescue targets is described below.

First, the Fire Brigade identifies potential rescue targets based on whether rescue is still possible and whether a sufficient number of agents are available. The time until rescue completion is considered for rescue agents, and for civilians, the time until transport to a refuge is considered. This prevents the death of rescue targets during the rescue process. Then, from the potential rescue targets, the Fire Brigade selects a target based on the priorities listed in Table 4.

Table 4: Priority of conditions used by our Fire Brigade

Priority	Criterion	Details
1	Target Type	Fire Brigade, Police Force, Ambulance Team, then Civilians
2	Path Distance	Shorter distance from the agent to the building containing the target
3	Buriedness	Lower buriedness (when multiple targets are in the same building)
4	Agent ID	Lower Agent ID (when all other criteria are equal)

The Fire Brigade prioritizes rescuing other rescue agents. This increases the number of active rescue agents, enabling more efficient civilian rescue. Prioritizing the Fire Brigade is particularly important because their ability to revive other rescue agents is essential for overall rescue efficiency.

Furthermore, the Fire Brigade prioritizes rescuing nearby targets. If multiple targets of the same type are in the same building, those with lower buriedness are prioritized. This is because prioritizing agents requiring less rescue time increases the total number of agents rescued.

Finally, if target type, path distance, and buriedness are identical, the agent with the lowest Agent ID is selected to ensure a unique rescue target.

These strategies enable more efficient rescue operations by prioritizing disabled disaster relief agents and maximizing the number of active disaster relief agents participating in rescue activities.

5 Preliminary Results

6 Conclusions

As part of our research, AIT-Rescue created agents using the Response Threshold Method. These agents operate in RRS environments without communication. We then verified the effectiveness of these Response Threshold Method agents in such communication-restricted environments. The results indicated that they tended to outperform Greedy Algorithm agents in some scenarios where agents were concentrated. However, they tended to underperform Greedy Algorithm agents in scenarios where agents were distributed or civilians were concentrated.

Future work includes investigating the relationship among information obtained from the environment and effective response thresholds. Our experiments confirmed that Response Threshold Method agents and Greedy Algorithm agents tend to excel in different scenarios. Additionally, the Response Threshold Method behaves similarly to the Greedy Algorithm when given a low response threshold. These results suggest the possibility of rescuing more civilians by dynamically adjusting the response threshold based on environmental information.

Also, AIT-Rescue is developing a new strategy for the police force for this year's competition. Specifically, the strategy is to perform debris cleaning along priority roads that are frequently used for transport. This is expected to secure transport routes from various locations early on. However, since it is currently in the development phase, its effectiveness has not yet been verified. Therefore, in the future, we plan to quantitatively evaluate its effectiveness through preliminary experiments and conduct comparative experiments with existing strategies.

Additionally, this year we will migrate our agents to ADF-Python. Therefore, we need to rewrite many of our existing Java modules as Python modules.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP21K12039 and JP23K11225.

References

1. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press (10 1999). https://doi.org/10.1093/oso/9780195131581.001.0001, `https://doi.org/10.1093/oso/9780195131581.001.0001`
2. Visser, A., Ito, N., Kleiner, A.: Robocup rescue simulation innovation strategy. In: Bianchi, R.A.C., Akin, H.L., Ramamoorthy, S., Sugiura, K. (eds.) RoboCup 2014: Robot World Cup XVIII. pp. 661–672. Springer International Publishing, Cham (2015)