RoboCup Rescue 2022
TDP Agent Simulation
Ri-one (Japan)

Takafumi Horie*, Takatsugu Nakayama, Kosuke Yano*, Haruto Nomoto, Shunnichi Kato, and Takuto Ikegami

Ritsumeikan University, Japan
rione.rescue@gmail.com
https://sites.google.com/view/ri-one

Abstract. We developed the system *Dynamic Task Management* which is implemented in Ri-one 2021. We implemented the method of integrating two indices of evaluation of deciding target: *Priority* and *Unstableness*, we realized flexible calculation. In ambulance teams (ATs), AT's module takes into capacity of refuges into account. We calculated waiting time for civilians to get treated at the refuge by using queuing theory and *Approximate Manhattan Distance*. AT's module compared civilian's HP at the time being observed and damage until waiting time for civilians to get treated at each refuge, and realized responses adopted to changes in Robocup 2022. As a result, the score increased compared to the implementation in 2021.

1 Introduction

One of the most important issues in Robocup Rescue Simulation (RRS) is the flexible management of tasks by agents. Improving a single module alone is not enough to solve this problem. For evaluation and selection of appropriate targets, the following are required; accurate behavior in Action, flexibly evaluation criterion in Detector, and integration of the results through all evaluation criteria. Section 2 explains improvements from last year's implementation. Section 3 explains the new strategy based on changes of rule in Robocup 2022.

2 Task Decision Framework: Dynamic Task Manager

2.1 Dynamic Task Management

2.1.1 Purpose Ri-one 2020 developed the system called *Dynamic Task Management*[2]. *Dynamic Task Management* is the system for flexible evaluation by separating the evaluation of entities and deciding the targets in the Detector modules. Firstly, all EntityIDs of entities are given to TaskManager class. Then, these all EntityIDs are allocated *Priority* and *Unstableness*. *Priority* indicates how much entities should be chosen as the targets. *Unstableness* indicates how much *Priority* of entities should be re-valuated. Secondly, revaluation and deciding the target are calculated in each cycle of execution. In revaluation, *Priority* and *Unstableness* of candidate that has the highest value of *Unstableness* are recalculated and updated its score of *Priority* and *Unstableness*. In order to reduce the computational complexity, recalculation of *Priority* and *Unstableness* is performed for limited candidates. When deciding the target, the entity that has

* Corresponding author.

the highest value of *Unstableness* is chosen, and given to the agent as the target by *Dynamic Task Management*. The detailed algorithm is shown in Fig. [2]

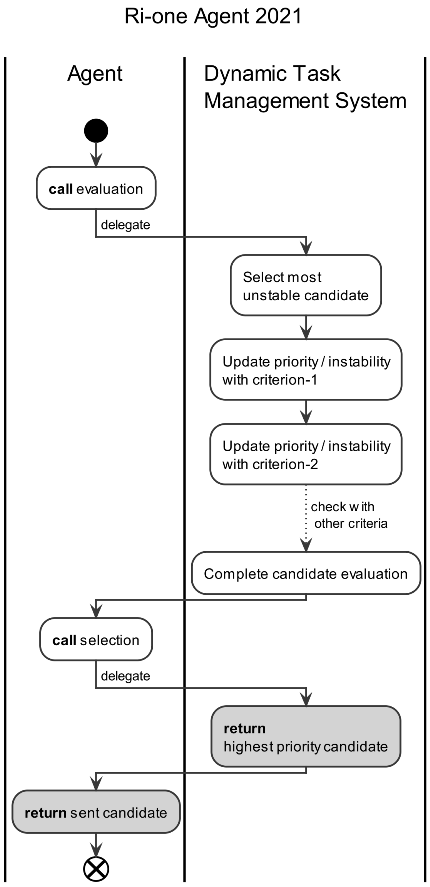

Fig. 1: Algorithm of *Dynamic Task Management*, [2]

Score was improved by *Dynamic Task Management*.[2] However, *Dynamic Task Management* has problems with deciding agents' tasks. *Dynamic Task Management* calculates the *Priority* and *Unstableness* for all candidates of target. In Ri-one2021, *Priority* and *Unstableness* in each criteria are considered to have same weight of importance, so *Dynamic Task Management* uses simple linear sum to calculate integrating *Priority* and *Unstableness*. However, weight of importance of criteria seems to change depending on maps and situations. Ri-one2022 implemented the improved method which takes situations of the disaster into account.

2.1.2 Caluculation Based on The Law of Propagation of Error In Ri-one2022, it is adopted two hypotheses for implementation of *Dynamic Task Management*. *Priority* and *Unstableness* are hereinafter defined as values of entities. First hypothesis is derived in the below formula.

$$P = \sum_k p_k \log(n_k + 2) \tag{1}$$

k means criteria and p_k means *Priority* in k. The definition area of p_k is up to 1 from 0. n_k means the number of targets that agents observed before calculation, and whose value of p_k is over 0.5. Therefore, this hypothesis assumes that events observed many times have high *Priority.* Second hypothesis is P and p_k include numerical error generate from normal distribution when they are calculated. In this hypothesis, the mean of the distribution generates from p_k is unobservable value μ_k. In this hypothesis, the variance of the distribution is shown in the below formula with the number of elapsed cycles t from observation by agents, and *Unstableness* σ_k calculated by each criterion k.

$$P \sim \mathcal{N}(\overline{P}, U^2) \tag{2}$$

In addition, the mean of the distribution conformed by P is thought to be unobservable value $\bar{P}$, and the variance of the distribution generates from P is thought to be U.

$$P \sim \mathcal{N}(\overline{P}, U^2) \tag{3}$$

From these formulas, observed value p_k and P are derived in the below formulas.

$$p_k = \mu_k \pm \sigma_k \log(t + 2) \tag{4}$$

$$P = \overline{P} \pm U \tag{5}$$

Integrated *Priority* $\overline{P}$ is simply derived in the below formula.

$$\overline{P} = \sum_k \mu_k \log(n_k + 2) \tag{6}$$

On the other hand, integrated *Unstableness* can't be calculated unlike $\overline{P}$ because U is the error. However, by using Propagation of Error[1], the error of μ_k is $\delta\mu_k = \sigma_k \log(t+2)$ i.e. U is calculated by the below formula.

$$U^2 = \sum_k \left(\frac{d}{d\mu_t}\overline{P}\right)^2 \cdot \delta\mu_k^2 \tag{7}$$

$$= \sum_k (\log(n_k + 2) \cdot \sigma_k \log(t + 2))^2 \qquad (\because \delta\mu_k = \sigma_k \log(t + 2)) \tag{8}$$

$$U = \sqrt{\sum_k ((\log(n_k + 2))^2 \cdot \sigma_k^2 (\log(t + 2))^2)} \tag{9}$$

Therefore, integrated *Priority* and integrated *Unstableness* are calculated through the formula (6) and (9).

2.1.3 Application to Other Modules Ri-one2021 introduced *Dynamic Task Management* to develop Detector modules[2]. Furthermore, in Ri-one2022, *Dynamic Task Management* is applied to Action modules on ATs. *Dynamic Task Management* is used to decide refuge that ATs transport the injured agent. In this application, The independence of modules are ensured by integrating TaskManager class to each Detector and Action modules as private fields. We will show concrete strategy in AT section.

2.1.4 Limitation In Ri-one2022, *Dynamic Task Management* is introduced in only Detector modules and AT's modules. Therefore, introducing *Dynamic Task Management* in other Action modules is expected in future implementation. In addition, the method of deciding tasks in *Dynamic Task Management* still has room for improvement. In the future outlook, it is predicted that adoption of machine learning like Online Learning to *Dynamic Task Management.*

3 Strategies

3.1 Ambulance Team

3.1.1 Purpose The purpose of Ri-one 2022's AT Module is to take refuges into account and to prevent civilians from death during waiting in line at refuge. *Dynamic Task Management* chooses refuges which maximize subtraction of civilian's HP at the time of obsevation and damage until waiting time for civilians to get treated.

3.1.2 Proposed Approach: Comparison of Time In refuge k, T_k means sum of waiting time for civilians to get treated and the time for moving to refuge.D_t means damage for time t, and $\int_0^T D_t dt$ means a damage which civilians get until time t. AT Module decides the *Priority* by calculating the value of $H_0 - \int_0^{T_k} D_t dt$ with *Dynamic Task Management.* AT Module decides higher value of $H_0 - \int_0^{T_k} D_t dt$, higher *Priority.* If the value of subtraction is 0 or less than 0, AT Module returns 0 as the *Priority* in this strategy.

3.1.3 Calculation Based on Queuing Theory AT Module takes about time for civilians to get treated at each refuge into account and uses [3][4] to calculate waiting time. Damage of each civilian is service time in queuing theory because time of which each civilian uses beds changes depending on damage of each civilian. Number of beds at each refuge is number of counters in queuing theory. Therefore, waiting time at each refuge is calculated based on queuing theory. S means the bed capacity. $\frac{1}{\mu}$ means time required for treatment of civilians. λ means number of agents joining queues at refuges. The occupancy rate ρ is derived in the below formula.

$$\rho = \frac{a}{S} \quad \left(a = \frac{\lambda}{\mu} \right) \tag{10}$$

ρ shows degree of how refuge k crowded. According to queuing theory, waiting time at refuge is derived in the below formula using a.

$$W_p = \frac{\frac{a^S}{S!}\frac{S}{S-a}}{\sum_{n=0}^{S-1} \frac{a^n}{n!} + \frac{a^S}{S!}\frac{S}{S-a}} \cdot \frac{1}{\mu(S-a)} \tag{11}$$

3.1.4 Calculation Based on Approximate Manhattan Distance Moving time to each refuge is calculated approximately as number of spent steps going to each refuge. l means number of entities which agents passed. It is called *Approximate Manhattan Distance* in this article. T_m means moving time from the agent's position to the refuge. According to the specifications of RRS, agents can move as long as 7000 entities while one step. Therefore, moving time T_m is derived in the below formula.

$$T_m = \frac{l}{7000} \tag{12}$$

Formula (11) and (12) led each length of time until each civilian gets treatment T_k. The formula to calculate length of time is shown below.

$$T_k = W_p + T_m \tag{13}$$

$$= \frac{\frac{a^S}{S!}\frac{S}{S-a}}{\sum_{n=0}^{S-1}\frac{a^n}{n!} + \frac{a^S}{S!}\frac{S}{S-a}} \cdot \frac{1}{\mu(S-a)} + \frac{l}{7000} \tag{14}$$

3.1.5 Calculation of Damage From the specifications of RRS, D_t is shown in the below formula by *DamageBury* B_t, *DamageCollapse* C_t and *DamageFire* F_t.

$$B_t = 0.000035 \cdot (B_{t-1})^2 + B_{t-1} + 0.01 + 2n \tag{15}$$

$$C_t = 0.00025 \cdot (C_{t-1})^2 + C_{t-1} + 0.01 + 2n \tag{16}$$

$$F_t = 0.00025 \cdot (F_{t-1})^2 + F_{t-1} + 0.03 + 2n \tag{17}$$

$$D_t = Round(B_t + C_t + F_t) \tag{18}$$

In those formulas, n means a random number generated from a normal distribution whose mean is 0.1 and variance is 0.01. Thus, the change amount of $x(\in \{B_t, C_t, F_t\}$: Δx is defined as formula (20) with using the constant k, l.

$$\Delta x = \frac{dx}{dt} \tag{19}$$

$$= kx^2 + l \tag{20}$$

$x(t)$ is change amount of civilian's HP; $x(t)$ is damage at time t. Solving formula(20) as a differential equation, $x(t) > 0$ is self-evident,

$$\frac{dx}{dt} \cdot \frac{1}{kx^2 + l} = 1 \tag{21}$$

Integrating both sides of formula(21) by t,

$$\frac{1}{\sqrt{kl}} \arctan \sqrt{\frac{k}{l}} x = t + C \tag{22}$$

$$x = \sqrt{\frac{l}{k}} \tan(\sqrt{kl} \cdot t + C) \tag{23}$$

where C means integration constant. Therefore, total damage until t is calculated in the below formula.

$$\int_0^t x dt = \int_0^t \sqrt{\frac{l}{k}} \tan(\sqrt{kl} \cdot t + C) dt \tag{24}$$

From the fomula: $\int \tan u du = -\log|\cos u|$, by putting $u = \sqrt{kl} \cdot t + C$,

$$\int_0^t x dt = \int_C^{\sqrt{kl} \cdot t + C} \frac{1}{k} \tan u du \tag{25}$$

$$= -\frac{1}{k} \log \left| \frac{\cos(\sqrt{kl} \cdot t + C)}{\cos C} \right| \tag{26}$$

By putting $x = x_0$ when $t = 0$, C is decided by (23) as follows.

$$x_0 = \sqrt{\frac{l}{k}} \tan C \tag{27}$$

$$C = \arctan \sqrt{\frac{k}{l}} x_0 \tag{28}$$

Hence, damage for time 0 to t is calculated by formula (26) in the below formula;

$$\begin{aligned}
\int_0^t D_t dt &= \int_0^t Round(B_t + C_t + F_t) dt \\
&= -\frac{1}{0.000035} \log \left| \frac{\cos(\sqrt{0.00000035}t + C_B)}{\cos C_B} \right| \\
&\quad - \frac{1}{0.00025} \log \left| \frac{\cos(\sqrt{0.0000025}t + C_C)}{\cos C_C} \right| \\
&\quad - \frac{1}{0.00025} \log \left| \frac{\cos(\sqrt{0.0000075}t + C_F)}{\cos C_F} \right| \\
&= -\frac{200000}{7} \log \left| \cos \frac{\sqrt{35}}{1000} t - 0.01\sqrt{35} x_{B0} \sin \frac{\sqrt{35}}{1000} t \right| \\
&\quad - 4000 \log \left| \cos \frac{\sqrt{0.1}}{500} t - 0.5\sqrt{0.1} x_{C0} \sin \frac{\sqrt{0.1}}{500} t \right| \\
&\quad - 12000 \log \left| \cos \frac{\sqrt{0.3}}{500} t - \frac{0.5\sqrt{0.1}}{3} x_{F0} \sin \frac{\sqrt{0.3}}{500} t \right|
\end{aligned} \tag{29}$$

where $t = 0$ is time at observation, x_0 is damage at time of observation, and integration constant of each damage is calculated by formula (28). C_B, C_C and C_F mean integration constant corresponding to B_t, C_t, F_t. T_k means time until civilians get treatment. Damage which civilians get until time T_k is calculated with substituting $t = T_k$ in formula (29).

$$\begin{aligned}
H_0 - \int_0^{T_k} D_t dt = H_0 &+ \frac{200000}{7} \log \left| \cos \frac{\sqrt{35}}{1000} T_k - 0.01\sqrt{35} x_{B0} \sin \frac{\sqrt{35}}{1000} T_k \right| \\
&+ 4000 \log \left| \cos \frac{\sqrt{0.1}}{500} T_k - 0.5\sqrt{0.1} x_{C0} \sin \frac{\sqrt{0.1}}{500} T_k \right| \\
&+ 12000 \log \left| \cos \frac{\sqrt{0.3}}{500} T_k - \frac{0.5\sqrt{0.1}}{3} x_{F0} \sin \frac{\sqrt{0.3}}{500} T_k \right|
\end{aligned} \tag{30}$$

Priority is decided based on a value of formula (30).

3.1.6 Limitation In this strategy, *Priority* of refuges is 0 when the value of $H_0 - \int_0^{T_k} D_t dt$ is 0 or less. If H_0 is too low, all candidate refuges' *Priority* will be 0. In this case, ATs get stuck because no refuges are selected as the target. In order to prevent such situations, it is necessary to consider not only selecting refuge's *Priorities* but also other selecting *Priorities* of actions.

4 Preliminary Results

In this section, we describe the results of the experiments. These experiments aimed to confirm the effectiveness of implemented functions. In this experiments, Ri-one 2022 reflects only strategy in section 2.

4.1 Experiment Conditions

In the experiments, Ri-one 2021 and Ri-one 2022 were compared. Ri-one 2021 refers to the implementation of 2021[2]. Ri-one 2022 refers to the modified version of Ri-one 2021. The simulations were carried out 10 times for each condition. The specifications of the used computer followed the table below.

Table 1: specs of a computer

OS	Ubuntu 18.04
CPU	Intel Core-i9 10850K
Memory	DDR4-2666 32GB

4.2 Score and Analyze

The experiment results followed the table below.

Map	score	
	Ri-one 2021	Ri-one 2022
berlin	74.30623829	74.40636703
eindhoven	190.9300255	190.9300255
istanbul	167.9030384	168.0033591
joao	128.7045793	128.7046294
kobe	182.2381594	182.6409453
montreal	72.681544	73.1837055
sakae	163.528685	163.528685
sf	99.87155892	99.67174823
vc	252.6770178	252.7769661

These results showed the effectiveness of improvements in Ri-one2022. The differences between Ri-one2021 and Ri-one2022 are a little. However, *Dynamic Task Management*is introduced in only Detector modules and AT's modules in Ri-one2022. It described the reason of why the differences are a little. In addition, these results show that even partial implementation makes differences. Therefore, the implementations in Ri-one 2022 seemed to be effective.

References

1. Hayashi, S., Baba, R.: 計測における誤差解析入門 (An Introduction to Error Analysis in measurements), pp. 83–84. Tokyo Kagaku Doujin (2000), (Taylor, J.R. An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements. ASMSU/Spartans.4.Spartans Textbook. 1997. University Science Books.)
2. Horie, T., Takehara, K., Kondo, H., Nakayama, T., Yano, K.: Robocuprescue 2021-rescue simulation league. team description ri-one (japan) (2021)
3. Kimura, S.: 確率工学シリーズ 1 待ち行列の数理モデル (Series of Probabilistic Engineering 1: Mathematical Model Waiting Queue), pp. 30–34. Asakura Publishing (2016)
4. Suzuki, T.: 基礎数学選書 8 待ち行列 (Basic Mathematics Selection 8: Waiting Queue), pp. 66–72. SHOKABO (2016)