RoboCupRescue 2022
TDP Agent Simulation AIT-Rescue (Japan)

Hiroya Suzuki[1], Akira Hasegawa[1], Haruki Uehara[1], Joe Fujisawa[1], Itsuki Matsunaga[1], Ryoya Maeda[1], Yuki Shimada[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. We modeled the RoboCupRescue Simulation as a distributed constraint optimization problem and studied communication volume reduction in the Binary Max-Sum_ADVP algorithm. In our experiments, we reduced the communication volume by approximately 97% and confirmed the effectiveness of our approach. However, we cannot implement our ideas on our agents owing to communication limitations. We therefore improved our agents using voice communication. This improvement enables agents to propagate information about civilians to each other and to communicate important information in a timely manner through radio communications. As a result, our agents perform better than they did in 2021.

1 Introduction

The task assignment problem is an important issue for the RoboCupRescue Simulation (RRS). This problem can be expressed as a distributed constraint optimization problem (DCOP) in a multi-agent system. Furthermore, the DCOP can be solved using a DCOP algorithm.

We therefore modeled the task assignment problem of the RRS as a DCOP and applied the Binary Max-Sum_ADVP algorithm [4]. However, the DCOP algorithm cannot be executed without an ADF extension environment [2] in the RRS. Additionally, the bandwidth is set for each channel in communication on the RRS, and it is thus not possible to have a large communication volume. We therefore attempted to reduce the communication volume of Binary Max-Sum_ADVP. As a result of our efforts, we reduced the communication volume of Binary Max-Sum_ADVP by approximately 97%.

However, communication limitations in the RRS do not permit the application of the above mentioned research. We therefore improved on last year's agent by propagating information about civilians through voice communication. Through voice communication, we can avoid the bandwidth pressure of radio communication, allowing the transmission of important messages with sufficient bandwidth through radio communication. This implementation resulted in our agent's score improving from last year in six out of eight scenarios.

Chapter 2 explains the DCOP and Binary Max-Sum_ADVP, presents the method of reducing the communication volume of Binary Max-Sum_ADVP, and reports on an experiment comparing the communication volume before and after the volume reduction. Chapter 3 describes the modules used in AIT-Rescue 2022. Chapter 4 presents the strategy of each agent implemented in AIT-Rescue 2022. Chapter 5 reports on an experiment conducted to evaluate the effectiveness of the implemented strategy. Chapter 6 summarizes the TDP and discusses issues to be addressed before RoboCup 2022.

2 Scientific topic

This chapter describes the modeling of the RRS task assignment problem as a DCOP and the Binary Max-Sum_ADVP algorithm. It also describes the reduction of the communication volume in this algorithm and evaluates the effect of the reduction.

2.1 Distributed Constraint Optimization Problem

The distributed constraint optimization problem (DCOP) is the problem of determining the combination of variable values that minimizes the cost of the variables corresponding to the distributed agents, given the variables and the constraints between the variables [1].

The DCOP is defined by the following five elements. We also explain the modeling of the task assignment problem of the RRS on the DCOP, using the example of a Fire Brigade.

- $\mathbf{A} = \{a_1, \ldots, a_m\}$
  is a set of agents, where a_i is an agent. In the RRS, it represents the set of all Fire Brigades.
- $\mathbf{X} = \{x_1, \ldots, x_p\}$
  is a set of variables. However, $p \geqslant m$, where p is the number of variables and m is the number of agents. In the RRS, the variables represent which tasks are assigned to which Fire Brigades. In other words, each variable is in correspondence with one agent.
- $\mathbf{D} = \{D_1, \ldots, D_p\}$
  is a set of discrete ranges for a finite set of variables $\mathbf{X}$. D_i is the range of variable x_i. Each element of the range is called a candidate value for the variable and is represented by d. The combination of any candidate values is represented by σ, and the combination of all candidate values is represented by Σ. Σ is obtained by the following Eq. (1). Additionally, $\sigma \in \Sigma$.

  $$\Sigma = \prod_{i=1}^{p} D_i \tag{1}$$

  In the RRS, we consider a set of ranges that represent all the tasks that can be assigned to each Fire Brigade. This paper regards a task as extinguishing a fire in the previous rule and describes an example of Fire Brigades.

- $\mathbf{F} = \{f_1, \ldots, f_l\}$
  is a set of functions (called cost functions) that represent constraints between variables. A cost function is defined as $f_i \colon \bigtimes_{x_j \in \mathbf{X}^i} D_j \rightarrow \mathbb{R}$. In this formula, $\mathbf{X}^i$ is the set of variables whose constraint relation is denoted by f_i.
  In the RRS, we represent a set of cost functions f with a variable $x \in \mathbf{X}^f$, where the variable is a parameter for all Fire Brigades that can be assigned to a task. Therefore, each cost function corresponds to one task.
- $\alpha \colon \mathbf{X} \rightarrow \mathbf{A}$
  is a mapping function of variables and agents. It represents which agent corresponds to which variable.

The objective function $\mathbf{F}_g(\sigma)$ for optimization is defined by Eq. (2). σ_{f_i} is a combination of only the candidate values corresponding to the cost function f_i from σ.

$$\mathbf{F}_g(\sigma) = \sum_{i=1}^{k} f_i(\sigma_{f_i}) \tag{2}$$

The optimal combination σ^* that minimizes the cost of $\mathbf{F}_g(\sigma)$ is defined by Eq. (3).

$$\sigma^* = \operatorname*{argmin}_{\sigma \in \Sigma} \mathbf{F}_g(\sigma) \tag{3}$$

In this paper, the selection of an arbitrary variable $x \in \mathbf{X}$ for any candidate value $d \in D_x$ in its range is represented by an ordered pair $\langle x, d \rangle$.

2.2 Binary Max-Sum_ADVP

Binary Max-Sum_ADVP [4] is a derivative algorithm of Binary Max-Sum [3] with a modified working graph and message content that improve the solution quality. Binary Max-Sum_ADVP converts the factor graph in which Binary Max-Sum operates into a directed acyclic graph (DAG), as shown in Fig.1, thereby eliminating closed paths and guaranteeing convergence of the solution. Additionally, the incorporation of the evaluation value and the variable's chosen value in the message content guarantees that the solution at least does not worsen with an increasing number of message communications. In this algorithm, sending and receiving not only the evaluated value but also the variable value selected by the variable node is called value propagation.

In both a factor graph and a directed acyclic graph, variables in the DCOP definition are represented as variable nodes, and cost functions are represented as function nodes. Edges connect variable nodes and function nodes and represent assignable relationships. During the conversion of the graph, all variable nodes are divided into root and leaf node sets. Which variable nodes belong to the root node set and the leaf node set changes as the algorithm runs. In this paper, the set of all nodes is $\mathbf{V}$, the set of variable nodes is $\mathbf{X}$, the set of root nodes is $\mathbf{R_x} \subset \mathbf{X}$, and the set of leaf nodes is $\mathbf{X} \backslash \mathbf{R_x}$. The set of variable nodes has the same definition in the set of variables $\mathbf{X}$ in the definition of the DCOP, and it is thus represented by the same symbol. The edges of the DAG are connected from the variable nodes in the root node set to all function nodes, and from each function node to all variable nodes in the leaf node set, as shown in Fig.1.

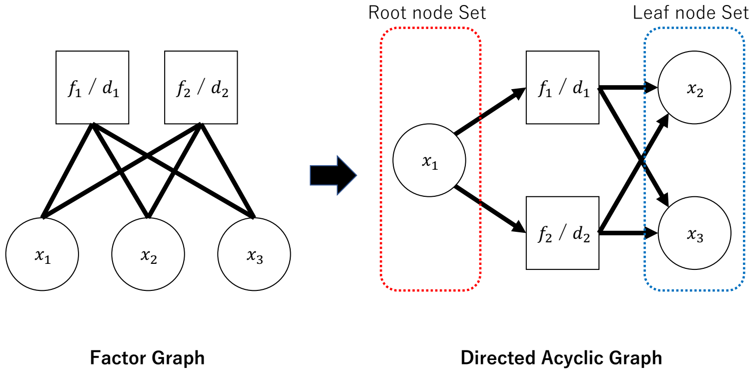

Fig. 1: Example of transforming a factor graph to a directed acyclic graph

In Binary Max-Sum_ADVP, all nodes work synchronously and repeatedly share the result with neighboring nodes in the graph. As a result, all nodes obtain solutions. This process is carried out in the following steps.

1. Receive messages from neighboring nodes.
2. Calculate the evaluation value based on the received information and generate a message with the evaluation value.
3. Send messages to neighboring nodes.

In the cycle of steps 1–3, the sending and receiving of a message is called a communication cycle. By repeating the communication cycle, the evaluation values calculated at each node are gradually transmitted to the entire graph. When the evaluation values of all nodes have been sufficiently transmitted to the entire graph, each variable node selects the candidate value with the highest evaluation value. Whether a message has been transmitted sufficiently is thus determined according to the status of change in the evaluation values of all nodes or by repeating a specified number propagations.

Binary Max-Sum_ADVP sends and receives messages according to the direction of the edge. In Fig.1, the variable node x_1 sends messages to the function nodes f_1, f_2, and the function nodes f_1, f_2 receive messages from the variable node x_1.

By repeating each of the three communication steps described above three times, it is possible to propagate messages between the most two distant nodes in the DAG. On the above basis, Binary Max-Sum_ADVP first executes the following process.

1. Form a DAG with 10% of all variable nodes in the root node set and the other variable nodes in the leaf node set.

2. Propagate messages through three communication cycles.
3. Set the current leaf node set as the root node set, reverse the edge orientation, and execute the process in step 2.

The stage in which these three steps are executed is called the AD phase.

In step 2 of the AD phase, the variable node generates a message using Eq. (4) and sends the message along the edge direction. Additionally, the function node generates a message using Eq. (5) and sends the message along the edge direction.

In step 2, the expression for obtaining the evaluation value to be sent from the variable node to the function node can be written as Eq. (4). Here, $\mathbf{N}_x$ represents the set of function nodes neighboring the variable node x, and f represents the destination function node.

$$m_{x\to f} = -\min_{f'\in\mathbf{N}_x\setminus\{f\}} m_{f'\to x} \tag{4}$$

In step 2, the expression for obtaining the evaluation value to be sent from the function node to the variable node is given in Eq. (5). Here, $\tilde{m}_{f_i\to x_j}(1)$ is the evaluation value when the variable node x_j of the transmission partner selects the value candidate d_i corresponding to f_i, and $\tilde{m}_{f_i\to x_j}(0)$ is the evaluation value when d_i is not selected. g_{ki} is a constant that represents the superiority or inferiority of a variable k to a value candidate d_i. $p_{f_i}(k)$ is a function that ranks how superior x_j is among the set of variable nodes $\mathbf{N}_{f_i}$ neighboring the function node f_i. Additionally, $p_{f_i}^{-1}(m)$ is the inverse function of $p_{f_i}(m)$.

$$m_{f_i\to x_j} = \tilde{m}_{f_i\to x_j}(1) - \tilde{m}_{f_i\to x_j}(0) + g_{ji} \tag{5}$$

where

$$\begin{aligned}
\tilde{m}_{f_i\to x_j}(b=\{1,0\}) &= \min_{n=0,\ldots,|\mathbf{N}_{f_i}|}\left(W^b_{f_i x_j,n} + E_{f_i x_{p_{f_i}^{-1}(m)},n}\right)\\
W^1_{f_i x_j,n} &= \begin{cases} w_i(n+1) & \text{if } n < p_{f_i}(j)\\ w_i(n) & \text{otherwise}\end{cases}\\
W^0_{f_i x_j,n} &= \begin{cases} w_i(n) & \text{if } n < p_{f_i}(j)\\ w_i(n-1) & \text{otherwise}\end{cases}\\
E_{f_i x_k,n} &= \sum_{\substack{m=1\\ m\neq p_{f_i}(j)}}^{n} e_{f_i x_{p_{f_i}^{-1}(m)}}\\
e_{f_i x_k} &= m_{x_k\to f_i} + g_{ki}\\
p_{f_i}(k) &= 1 + |\,\{x_r \mid x_r \in \mathbf{N}_{f_i},\\
&\qquad e_{f_i x_r} < e_{f_i x_k} \vee ((e_{f_i x_r} = e_{f_i x_k}) \wedge r < k)\}\,|
\end{aligned}$$

The value candidate $\sigma^*_{x_i}$ that the variable node x_i finally selects is calculated using Eq. (6).

$$\sigma^*_{x_i} = \underset{d_j \in D_i}{\text{argmin}}\ m_{f_j \to x_i} \tag{6}$$

After the AD phase is completed, the following processes are executed until the termination conditions are satisfied. In this paper, the termination condition is set at 300 communication cycles. This condition is a heuristic value. Marc et al. set it at 100 communication cycles. [3]

1. Generate a DAG with only the top-level variable nodes in the order relation $\prec_{\mathbf{X}}$ of the variable, as the root node set and the other variable nodes as the leaf node set.
2. Let the variable node added to the root node set select a value using Eq. (6).
3. Propagate messages through three communication cycles.
4. Execute the following processes according to the number of variable nodes included in the leaf node set.

   For two or more Add the topmost variable node in the variable ordering relation $\prec_{\mathbf{X}}$ to the root node set from the variable nodes included in the leaf node set.

   For one Make the current leaf node set the root node set, and reverse the ordering relation of variables $\prec_{\mathbf{X}}$ and nodes $\prec$ and the direction of edges.
5. Repeat steps 2 to 4 until the termination condition is satisfied.

This stage following the AD phase is called the VP phase. In step 3 of the VP phase, the variable node calculates the evaluation value using Eq.(4), as in the AD phase, and sends the messages including the evaluation value and selected value. The function node reflects the value received in the value propagation in the partial assignment $\widehat{\sigma_f}$ of the cost function f and sends a message using Eq. (7).

Here, $\prec_{\mathbf{X}}$ uses the ascending order of the index of the variable. The order relation of the nodes, $\prec$, is defined as follows. The direction of the edges on the DAG is decided by the order relation. Here, v_1 and v_2 represent any nodes, and X represents any variable node.

$$v_1 \prec v_2 \Longleftrightarrow \begin{cases} v_1 \prec_{\mathbf{X}} v_2 & v_1 \in \mathbf{R_x}, v_2 \in \mathbf{X}\backslash\mathbf{R_x} \\ v_1 \prec_{\mathbf{X}} X & v_1 \in \mathbf{R_x}, v_2 \in \mathbf{F}, X \in \mathbf{X}\backslash\mathbf{R_x}, \{v_1, X\} \subset \mathbf{N}_{v2} \\ X \prec_{\mathbf{X}} v_2 & X \in \mathbf{R_x}, v_1 \in \mathbf{F}, v_2 \in \mathbf{X}\backslash\mathbf{R_x}, \{X, v_2\} \subset \mathbf{N}_{v1} \end{cases}$$

The above definitions have the following meaning.

- The variable nodes in the root node set are higher in the order relation than the variable nodes included in the function node and leaf node sets.
- Function nodes are lower in the order relation than the variable nodes included in the root node set. In addition, function nodes are higher in the order relation than variable nodes included in the leaf node set.
- The variable nodes included in the leaf node set are lower in the order relation than the variable and function nodes included in the root node set.

An example of graph conversion in the VP phase is shown in Fig.2. In Fig. 2, the upper-left value is the number of iterations of steps 2 to 4 in the VP phase. In the fifth iteration, the graph is the same as that in the first iteration, and the graph is converted continuously in the same way.

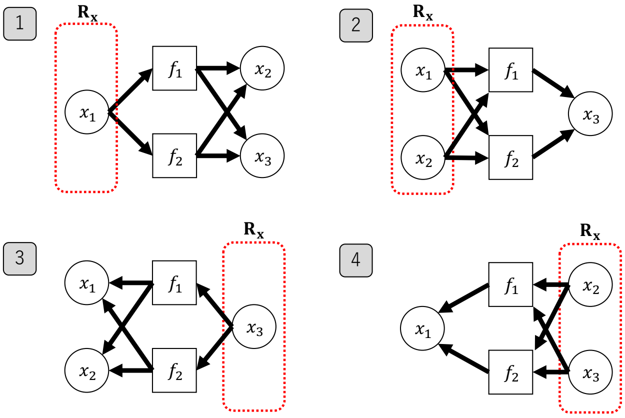

Fig. 2: Example of graph conversion in the VP phase

Equation (7) is used to calculate the message to be sent by the function node f_i to the variable node x_j belonging to the leaf node set in step 3 of the VP phase.

$$m_{f_i \to x_j} = \tilde{m}_{f_i \to x_j}(1) - \tilde{m}_{f_i \to x_j}(0) + g_{ji} \tag{7}$$

where

$$
\begin{aligned}
\tilde{m}_{f_i \to x_j}\left(b = \{1, 0\}\right) &= \min_{n = n_{f_i}^{d_i}, \ldots, |\mathbf{N}_{f_i}| - n_{f_i}^{-d_i}} \left(W^b_{f_i x_j, n} + E_{f_i x_{p_{f_i}^{-1}(m)}, n}\right) \\
W^1_{f_i x_j, n} &= \begin{cases} w_i(n+1) & \text{if } n < p_{f_i}\left(j\right) \\ w_i\left(n\right) & \text{otherwise} \end{cases} \\
W^0_{f_i x_j, n} &= \begin{cases} w_i(n) & \text{if } n < p_{f_i}\left(j\right) \\ w_i\left(n-1\right) & \text{otherwise} \end{cases} \\
E_{f_i x_k, n} &= \sum_{\substack{m=1 \\ m \neq p_{f_i}(j)}}^{n} e_{f_i x_{p_{f_i}^{-1}(m)}} \\
p_{f_i}\left(k\right) &= 1 + \mid \{x_r \mid x_r \in \mathbf{N}_{f_i}, \\
&\qquad (e_{f_i x_r} + \epsilon_{f_i}(x_r)) < (e_{f_i x_k} + \epsilon_{f_i}(x_k)) \\
&\qquad \vee ((e_{f_i x_r} = e_{f_i x_k}) \wedge r < k)\} \mid \\
e_{f_i x_k} &= m_{x_k \to f_i} + g_{ki} \\
n_{f_i}^{d_i} &= |\{\langle x, d\rangle \mid d = d_i, \langle x, d\rangle \in \widehat{\sigma_{f_i}}^{-x_j}\}| \\
n_{f_i}^{-d_i} &= |\{\langle x, d\rangle \mid d \neq d_i, \langle x, d\rangle \in \widehat{\sigma_{f_i}}^{-x_j}\}| \\
\epsilon_{f_i}(x_k) &= \begin{cases} -\epsilon & \text{if } \{\langle x, d\rangle \mid (x = x_k) \wedge (d = d_i), \langle x, d\rangle \in \widehat{\sigma_{f_i}}\} \neq \varnothing \\ \epsilon & \text{if } \{\langle x, d\rangle \mid (x = x_k) \wedge (d = d_i), \langle x, d\rangle \in \widehat{\sigma_{f_i}}\} = \varnothing \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

Here, in the expression for $\widehat{\sigma_{f_i}}^{-x_j}$, $n_{f_i}^{d_i}$ is the number of variable nodes that have selected the value candidate d_i corresponding to the cost function f_i whereas $n_{f_i}^{-d_i}$ is the number of variable nodes that have not selected d_i corresponding to f_i. Additionally, in $p_{f_i}(j)$, $\epsilon_{f_i}(x_j)$ is a function that changes the rank of the variable node $x_k \in \mathbf{N}_{\mathbf{f_i}}$ according to $\widehat{\sigma_{f_i}}$. This function changes the rank of the variable nodes in Eq. (7) so that the order is variable nodes that have selected d_i in $\widehat{\sigma_{f_i}}^{-x_j}$, variable nodes and destination variable nodes that are not included in $\widehat{\sigma_{f_i}}^{-x_j}$ at that time, and variable nodes that have not selected d_i in $\widehat{\sigma_{f_i}}^{-x_j}$.

Equation (7) determines the rank that can be obtained by $p_{f_i}(j)$ according to Eq. (7), and limits the range of indices of variable nodes calculated by $m_{f_i \to x_j}(1)$ and $m_{f_i \to x_j}(0)$ from $n_{f_i}^{d_i}$ to $\mid \mathbf{N}_{f_i} \mid -n_{f_i}^{-d_i}$, so that the values of variable nodes that select d_i in $\widehat{\sigma_{f_i}}^{-x_j}$ are always added, and the values of variable nodes that do not select d_i are never added. The values of variable nodes that have d_i selected in $\widehat{\sigma_{f_i}}^{-x_j}$ are always added, and the values of variable nodes that have not selected d_i are never added. This means that at each function node, it is possible to obtain the best evaluation value and its partial assignment at the destination variable nodes and the variable nodes not included in the partial assignment, after limiting the solution space using the partial assignment of that variable node.

2.3 Reduction of the communication volume for Binary Max-Sum_ADVP

The RRS has communication limitations. In radio communication, each channel has its own bandwidth, which limits the volume of messages that rescue agents can communicate. In voice communication, there is a limit to the range of messages that agents can send, and only one message can be communicated in one step. Binary Max-Sum_ADVP described in Section 2.2 requires much communication and it is thus important to reduce the communication volume.

In this paper, we propose the following method of message reduction.

Algorithm behavior and reduction using graph structures In step 3 of the VP phase described in Section 2.2, messages including new content are propagated in the order of the root node set, function node set, and leaf node set as shown in Fig. 3. The reason is that each node calculates the new message to be sent on the basis of the messages immediately before the node received. The contents of these sent messages is then reflected on each node after one communication cycle because the nodes operate synchronously.

Each node calculates the evaluation value to send according to the evaluation value received in the previous communication cycle. In other words, if a message has no new information, no node needs to send the message. Thus, each node stops sending messages below in each communication cycle.

First communication cycle Messages sent by variable nodes that have not been added to the root node set and messages from function nodes

Second communication cycle Messages from variable nodes belonging to the root node set

Third communication cycle Messages from all nodes

Reduction using algorithmic behavior In the communication cycle immediately before a variable node is added to the root node set, only the highest variable node in $\prec_{\mathbf{X}}$ among the variable nodes in the leaf node set needs to receive a message from the function node. The variable nodes included in the leaf node set do not send messages because they have no destination node. Such nodes do not send and they thus do not need to calculate or receive an evaluation value. However, in the case of a variable node that moves to the root node set in the next communication cycle, the node needs to receive the evaluation value and calculate the value.

Therefore, the function nodes stop sending messages to the variable nodes belonging to the leaf node set, except for the highest variable node in $\prec_{\mathbf{X}}$.

Reduction through message comparison For the same reason as in the reduction using the algorithm behavior and graph structure, if the messages are

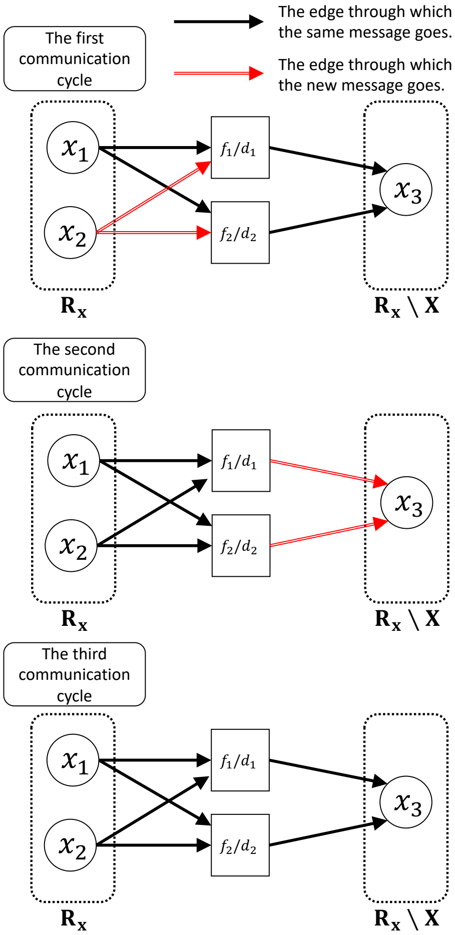
The edge through which the same message goes.
The edge through which the new message goes.
The first communication cycle
The second communication cycle
The third communication cycle
x_1
x_2
x_3
f_1/d_1
f_2/d_2
$\mathbf{R_x}$
$\mathbf{R_x \setminus X}$

Fig. 3: Messages containing new information for each communication cycle in step 2 of the VP phase

the same as in the previous communication cycle, nodes do not need to send the messages because such messages do not affect the calculation of the evaluation value or the assignment. Therefore, each node compares the message it sent in the previous communication cycle with the message it plans to send in the current communication cycle, and stops sending if they are the same.

2.4 Experiment

This experiment compares the communication volume of Binary Max-Sum_ADVP before the reduction with that after the reduction through simulations with various conditions. In this experiment, we use the ADF extension environment for

DCOP [2]. In other words, each agent can communicate multiple times during one step.

We use the VC2 and SF2 map of RoboCup 2021 and prepare a total of 480 disaster scenarios for experiments. The average numbers of messages before and after the communication volume reduction are compared in Table 1. The experiment uses the initial step of each simulation, and the algorithms run 300 cycles during the step. In other words, we focus on the initial condition of the disaster and the simulation state does not change with time.

Table 1: Comparison of the number of messages

Number of messages	Minimum	Average	Maximum	Standard deviation
Before reduction	15,000	15,000	15,000	0
After reduction	460	497	886	31

As shown in Table 1, the average number of messages after the communication volume reduction is 497, whereas the average number of messages before the reduction is 15,000. Therefore, the method of reducing the volume of communication proposed in this TDP reduces the communication volume by approximately 97% on average. There is no change in the solution before and after the reduction of the communication volume in any disaster scenario.

3 Modules

3.1 Refuge Selector

Since last year, the refuge capacity was introduced to the RRS. Having a capacity means that the refuges admit only a limited number of victims to their premises. This change affects how the Ambulance Teams decide where to carry victims. The agents get the following information about the refuge capacity.

- bedCapacity: total number of beds available in a refuge
- occupiedBeds: number of occupied beds in a refuge
- waitingListSize: number of victims already waiting in a queue for treatment in a refuge

We therefore developed Refuge Selector. Refuge Selector is a module that decides to which refuge the Ambulance Teams transport victims. Refuge Selector calculates for each refuge the utility of selecting this refuge. The utility for each refuge is shown in Eq. (8). Refuge Selector decides the refuges where this utility is maximized.

$$utilityVariable = (maxDistance - distanceR_i) + \left(\frac{bedCapacityR_i}{refugeSize \times 2}\right) \quad (8)$$

where $maxDistance$ is the longest straight-line distance between an Ambulance Team and any refuge, $distanceR_i$ is the straight line distance between an

Ambulance Team and a refuge R_i, $bedCapacityR_i$ is a bed capacity of a refuge R_i, and $refugeSize$ is the number of refuges on the map. $bedCapacityR_i$ is divided by $refugeSize \times 2$, which is currently an empirically determined value, meaning that it is simply a coefficient.

It is necessary to consider a utility formula that takes into account occupiedbeds and waitingListSize. However, Ambulance Teams never have the latest information, and the reliability of the information is low. Therefore, in this paper, we do not use these two pieces of information.

3.2 Overcrowding Area (Clustering)

In some disaster scenarios, agents may initially deploy densely. In addition, in the RRS, agents cannot move when blocked by blockades and cannot commence rescue tasks quickly. Therefore, the Police Force must remove the blockades in areas with high agent density as soon as possible.

We developed a clustering module called Overcrowding Area. Overcrowding Area is a module that detects the area where the Police Force, Ambulance Teams, and Fire Brigades are densely packed.

When the same type of agent exists in the perceptual range of each agent for more than half of the total number of agents, Overcrowding Area holds EntityID of the road or building where the agents are located.

4 Strategies

4.1 Ambulance Team

An Ambulance Team is responsible for transporting civilians who have been rescued by a Fire Brigade to refuges. This implies that cooperation between an Ambulance Team and Fire Brigades is essential.

Cooperation with other agents In the 2021 TDP [5], when some agents found a civilian who was buried in a collapsed building, the Ambulance Team used radio communications to communicate with the Fire Brigades about the buried civilian and to request rescue. However, radio communication has a bandwidth cap, and it is not easy to provide such communication stably. Therefore, the Ambulance Team adopts a strategy whereby agents mainly use voice communications within their communicable range. The strategy is that agents send rescue requests and propagate information about civilians to other agents.

Using voice communications avoids bandwidth pressure on radio communications and communicates critical information with plenty of bandwidth. The following critical case requires radio communications. If the rescue agents cannot move, radio communications are effective because of the delay in rescuing victims. Agents need to send requests to the Police Force if a road is blocked or a Fire Brigade if a victim is buried in rubble.

Search and transport The Ambulance Team adopts merged clustering as described in the 2021 TDP [5]. Merged clustering assigns an area on a map to a properly responsible agent. The area is called a cluster. The cluster assigned to each agent can be extended by merging neighboring clusters. In the case that the agent has no more tasks to complete in the assigned area, the agent can search for new tasks in other areas.

The Refuge Selector described in Section 3.1 determines the transport destination.

4.2 Fire Brigade

A Fire Brigade is responsible for rescuing the buried civilians and readying them for transportation by the Ambulance Team. Therefore, a Fire Brigade must cooperate with the Ambulance Team.

Cooperation with other agents In the 2021 TDP [5], the Fire Brigade sent information of the targeted civilian and a transportation request to the Ambulance Team via communication immediately after agents finished rescuing a buried civilian. Thus, the Fire Brigade does not use radio communication but voice communication to rescue civilians as described by the Ambulance Team.

The Fire Brigade adopts the same voice communication strategy as the Ambulance Team to ensure that there is sufficient bandwidth to communicate the critical information.

Rescue operations prioritize the number of surviving agents and civilians The Fire Brigade rescues rescue agents and civilians according to the priorities in Table 2. Assigning higher priority to the rescue agents improves the efficiency of rescuing civilians.

Table 2: Priority of conditions used by our Fire Brigade teams

Priority	Condition
1	Agent in the cluster
2	Civilian in the cluster
3	Agent outside of the cluster
4	Civilian outside of the cluster

The Fire Brigade prioritizes a life-savable civilian as much as possible. In other words, the Fire Brigade decides whether it rescues a found buried civilian according to Eq. (9). *timeToReach* is the number of steps from the current location to the nearest refuge.

$$(timeToReach + buriedness) * damage) < hp \tag{9}$$

4.3 Police Force

The Police Force is responsible for clearing debris from a blocked road and allowing other agents to move along the calculated path. The rescue operation will be delayed if a rescue agent cannot move owing to the blockage of roads.

Therefore, if rescue agents are initially concentrated at certain locations, the Police Force uses the clustering module described in Section 3.2 to remove debris from the dense areas with the highest priority. Table 3 gives the priority of the Police Force in clearing debris. In the case of the same priority, the Police Force firstly clears the debris at the closest linear distance from the current position.

Table 3: Priority of conditions used by our Police Force

priority	target
1	Refuges in its assigned cluster
2	Roads and buildings in an area where many agents are initially deployed and concentrated
3	•Ambulance Teams and Fire Brigades stuck in rubble within perceptual range •Debris on a path which a rescue agent moves along within perceptual range
4	Requests for clearing a blockage from an Ambulance Team or a Fire Brigade
5	Buildings where a rescue agent is initially deployed, in its assigned cluster
6	Highways in its assigned cluster
7	Buildings in its assigned cluster
8	Roads in its assigned cluster

Cooperation with other agents The Police Force receives a request from the Ambulance Team and the Fire Brigade to clear the blockage and works to rescue agents stuck in the rubble. If rescue agents are within the range of voice communication, they will send a request and the information of found civilians via voice communication.

5 Preliminary Results

We conducted comparison experiments using our agent from last year to examine the effectiveness of AIT-Rescue 2022. The scenarios were taken from the RoboCup 2021 final scenarios. The experimental results are given in Table 4.

The experimental results show that our improved agent scored higher than our agent from last year in six out of eight scenarios. In addition, the scores improved greatly for the berlin3 and Kobe3 scenarios. The higher scores were due to the scenario conditions being suitable for our strategies and modules. However, the map sizes of berlin3 and Kobe3 are different, so that we will clarify the reason in the future.

The score dropped by approximately 0.1 for the paris3 scenario. The rescue request was made via radio communication in AIT-Rescue2021 and via voice communication in AIT-Rescue 2022. There was thus a delay in the communication of information and the agent's rescue operation in AIT-Rescue2021.

Table 4: AIT-Rescue 2021 and AIT-Rescue 2022 Experimental results

Scenario	**Team**	
	AIT-Rescue 2021	**AIT-Rescue 2022**
berlin3	12.760	22.063
eindhoven3	38.479	40.999
kobe3	19.499	27.721
montreal1	40.384	41.576
paris3	66.949	66.830
sakae2	53.446	54.972
sf2	17.748	18.463
sydney2	4.707	4.707

The above assumes that the modules and strategies that we have described in this paper work effectively in the RRS.

6 Conclusions

We modeled part of the RRS task assignment problem as a DCOP and applied a DCOP algorithm, Binary Max-Sum_ADVP. In addition, we attempted to reduce the communication volume of Binary Max-Sum_ADVP for use in competitions. As a result, the communication volume was reduced by approximately 97%. However, this algorithm cannot yet be used in competition because it requires an ADF extended environment [2].

We therefore implemented a method of propagating information about rescuing civilians via voice communication. The voice communication strategy avoids the bandwidth pressure of radio communication and transmits critical communication with plenty of bandwidth via radio communication. Our agent surpassed its scores achieved last year in six out of eight scenarios through the introduction of our method.

For RoboCup 2022, we will further improve Eq. (9), as shown in Section 4.2. The damage done to the citizens is added to each step in the RRS. Additionally, Eq. (9) does not consider the time it takes for the Ambulance Team to get to the location of the civilians. These problems introduce difficulty in making accurate predictions. We will therefore improve the equations used in predictions and apply them to our agent.

References

1. Fioretto, F., Pontelli, E., Yeoh, W.: Distributed constraint optimization problems and applications: A survey. CoRR **abs/1602.06347** (2016), `http://dblp.uni-trier.de/db/journals/corr/corr1602.html#FiorettoPO16`
2. Miyamoto, Y., Kusaka, T., Okado, Y., Iwata, K., Ito, N.: RoboCupRescue 2019 TDP Infrastructure AIT-Rescue (Japan) (2019)

3. Pujol-Gonzalez, M., Cerquides, J., Farinelli, A., Meseguer, P., Rodríguez-Aguilar, J.A.: Binary max-sum for multi-team task allocation in RoboCup Rescue. In: Optimisation in Multi-Agent Systems and Distributed Constraint Reasoning (OptMAS-DCR) (2014)
4. Toshinari, S., Yuki, O., Yuki, M., Taishun, K., Takeshi, U., Kazunori, I., Nobuhiro, I.: RDevelopment of a DCOP Pseudo-Communication Module for RRS and Improvement of Binary Max-Sum. Proceedings of Fuzzy System Symposium **37**, 501–506 (sep 2021), `https://ci.nii.ac.jp/naid/40022697127/`
5. Yuki, O., Toshinari, S., Akira, H., Hiroya, S., Haruki, U., Kazunori, I., Nobuhiro, I.: RoboCupRescue 2021 TDP Agent Simulation AIT-Rescue (Japan) (2021)