AIT-Rescue

Yuki Miyamoto, Taishun Kusaka, Yuki Okado
Kazunori Iwata, Nobuhiro Ito

Aichi Institute of Technology / Aichi University

Agenda

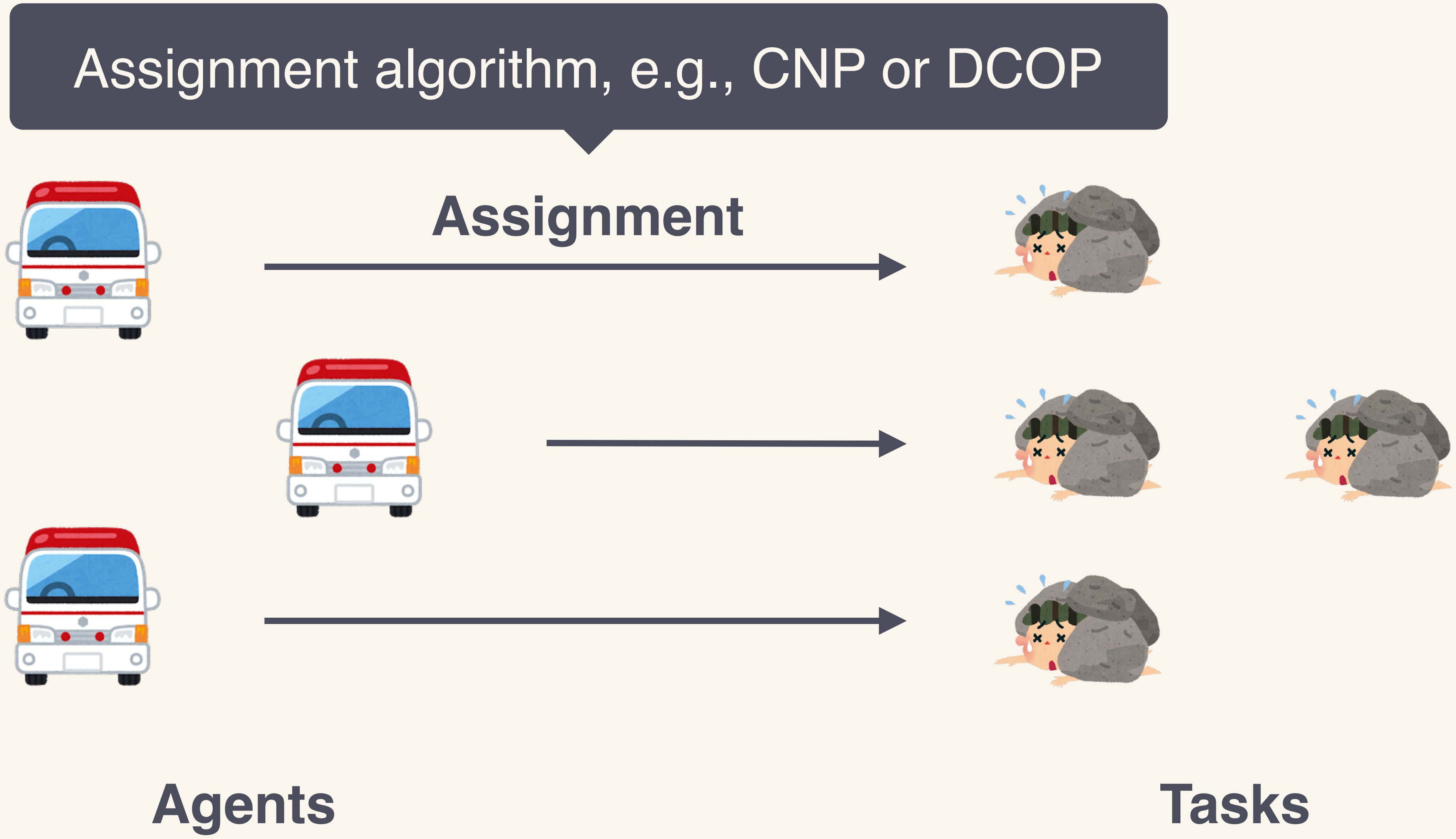
① The DCOP framework is useful for solving task assignment problems.

② We propose an environment that makes it possible to use DCOP on RoboCupRescue Simulation (RRS).

③ We demonstrate simulations using our proposed environment & show their results.

Task Assignment Problem

- A typical problem in RRS.
- It is finding an approach to assign *n* agents to *m* tasks.

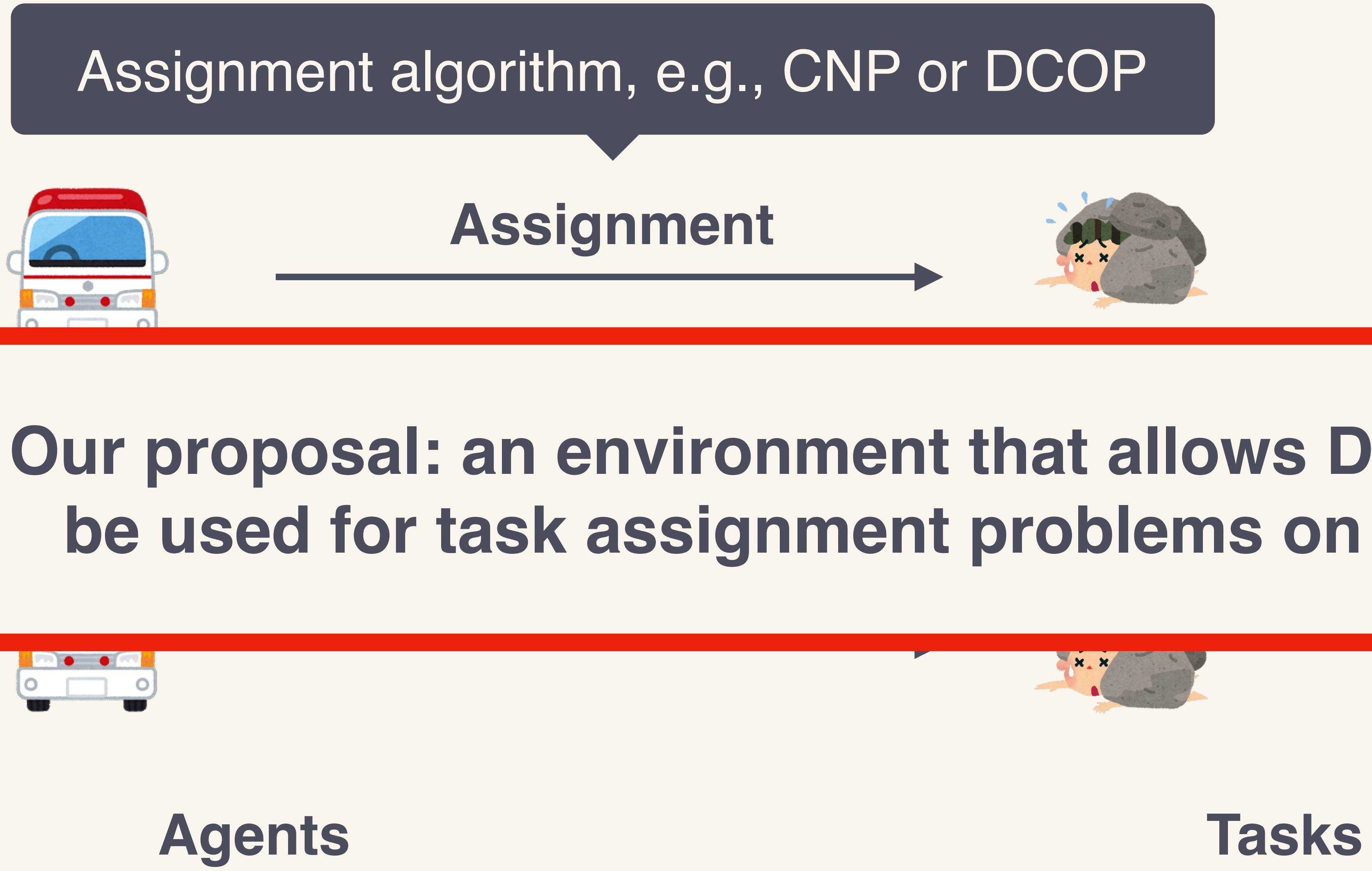


Assignment algorithm, e.g., CNP or DCOP

Task Assignment Problem

- A typical pr
- It is finding

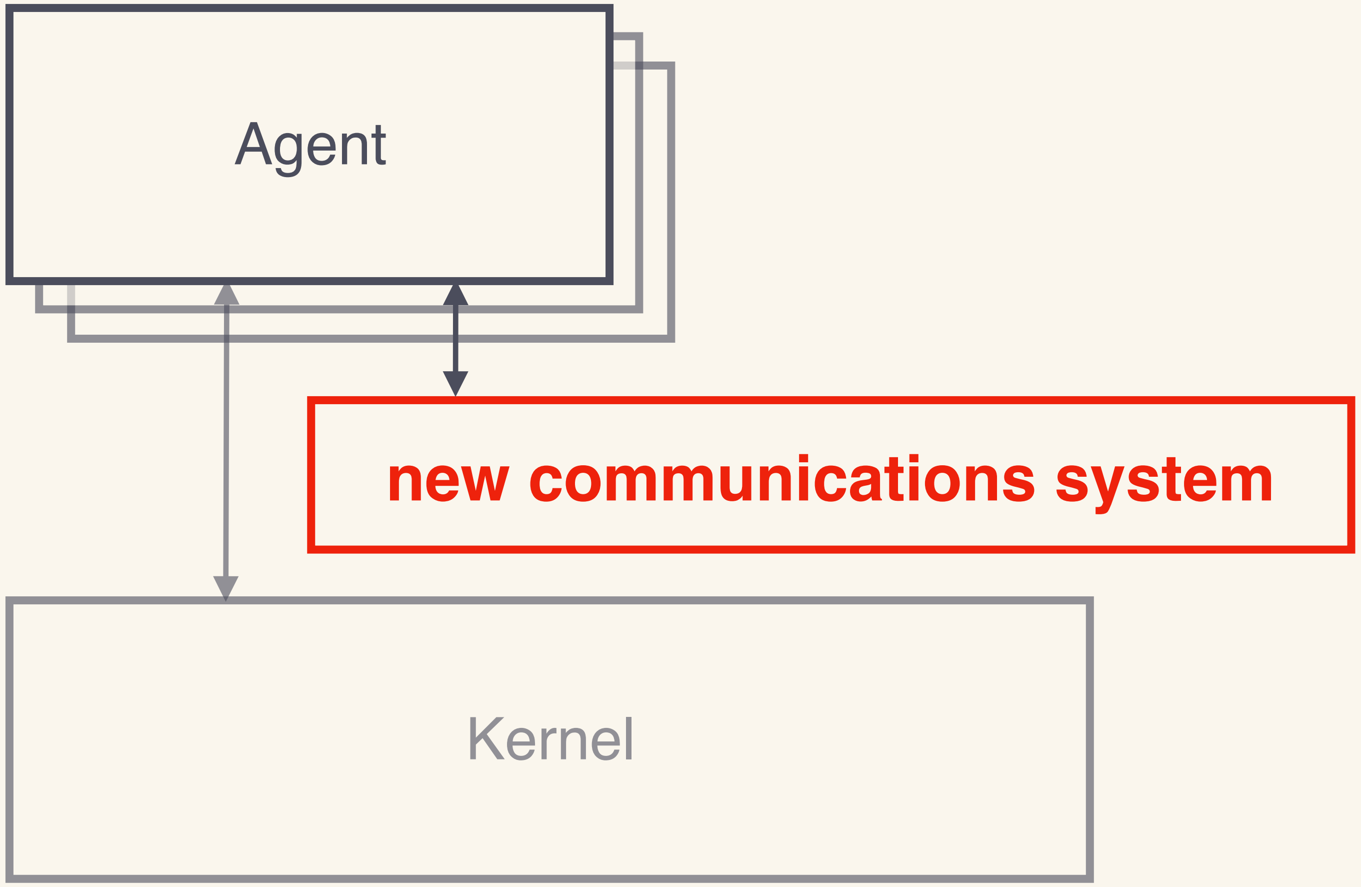
DCOP attracted attentions at AAAI-18.
→ DCOP is standard on Multi-Agent


Assignment algorithm, e.g., CNP or DCOP

Our proposal: an environment that allows DCOP to be used for task assignment problems on RRS.

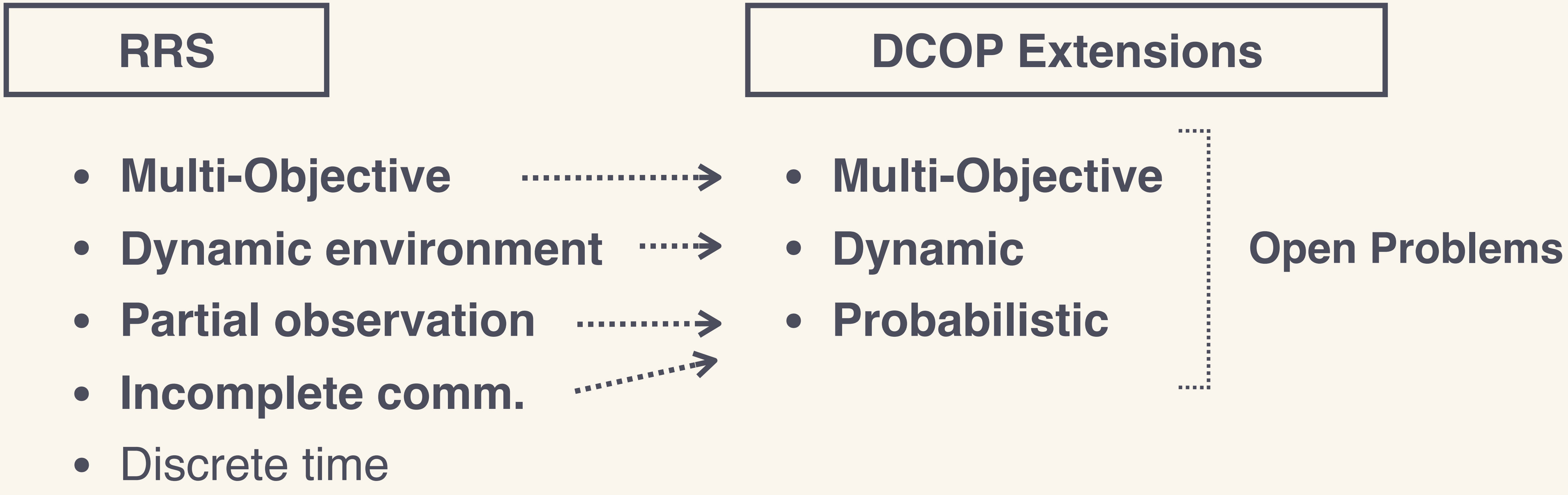
Task Assignment Problem

If we use a new communications system
on the RRS system to allow the use of the DCOP...



Importance of DCOP for RRS

- Task assignment problem features on RRS can be modeled as an extended DCOP.



RRS

- Multi-Objective
- Dynamic environment
- Partial observation
- Incomplete comm.
- Discrete time

DCOP Extensions

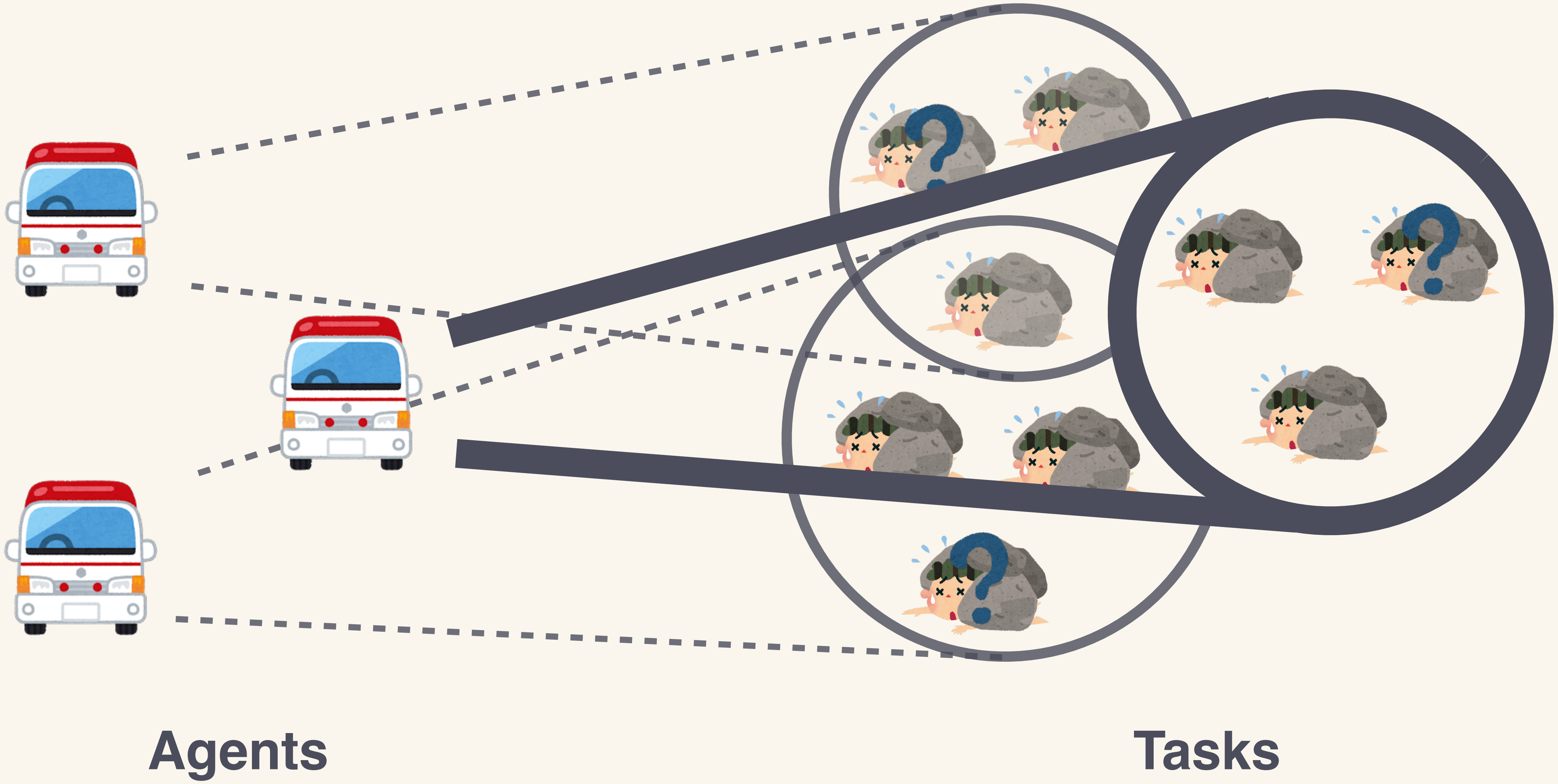
- Multi-Objective
- Dynamic
- Probabilistic

Open Problems

We can research new DCOP algorithm by combining these extensions on RRS.

Task Assignment Problem (Decentralized)

- When each agent performs task assignment individually, each agent selects a task from tasks in each scope.
- These kinds of problems can be modeled precisely as DCOP.



Definition of DCOP

Distributed Constraint Optimization Problem

- The problem of determining a combination of variable values that maximize utility.

Formalization

F. Fioretto, E. Pontelli, and W. Yeoh: "Distributed Constraint Optimization Problems and Applications: A Survey"

$$DCOP = < \mathbf{A}, \mathbf{X}, \mathbf{D}, \mathbf{F}, \alpha >$$

$\mathbf{A} = \{ a_1, \ldots, a_n \}$: a set of agents.

$\mathbf{X} = \{ x_1, \ldots, x_m \}$: a set of variables.

$\mathbf{D} = \{ D_1, \ldots, D_m \}$: a set of ranges for each variable.

$\mathbf{F} = \{ f_1, \ldots, f_k \}$: a set of utility functions (constraints).

$\alpha : \mathbf{X} \rightarrow \mathbf{A}$: a mapping function.

Definition of DCOP

Distributed Constraint Optimization Problem

- The problem of determining a combination of variable values that maximize utility.

Formalization

F. Fioretto, E. Pontelli, and W. Yeoh: "Distributed Constraint Optimization Problems and Applications: A Survey"

$$DCOP = < \mathbf{A}, \mathbf{X}, \mathbf{D}, \mathbf{F}, \alpha >$$

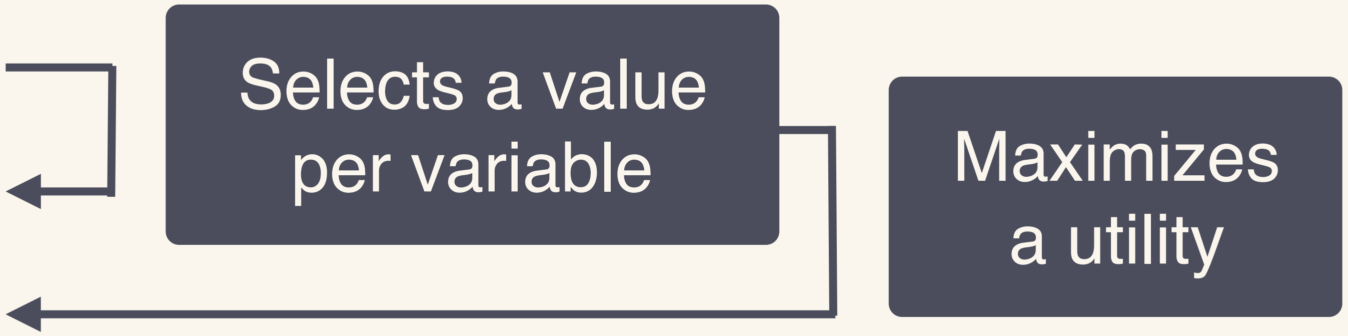
$$\mathbf{A} = \{ a_1, \ldots, a_n \}$$

$$\mathbf{X} = \{ x_1, \ldots, x_m \}$$

$$\mathbf{D} = \{ D_1, \ldots, D_m \}$$

$$\mathbf{F} = \{ f_1, \ldots, f_k \}$$

$$\alpha : \mathbf{X} \to \mathbf{A}$$



Selects a value per variable

Maximizes a utility

Definition of DCOP

Distributed Constraint Optimization Problem

- The problem of determining a combination of variable values that maximize utility.

Formalization

F. Fioretto, E. Pontelli, and W. Yeoh: "Distributed Constraint Optimization Problems and Applications: A Survey"

$$DCOP = < \mathbf{A}, \mathbf{X}, \mathbf{D}, \mathbf{F}, \alpha >$$

$$\mathbf{A} = \{ a_1 , \ldots, a_n \}$$

$$\mathbf{X} = \{ x_1 , \ldots, x_m \}$$

$$\mathbf{D} = \{ D_1 , \ldots, D_m \}$$

$$\mathbf{F} = \{ f_1 , \ldots, f_k \}$$

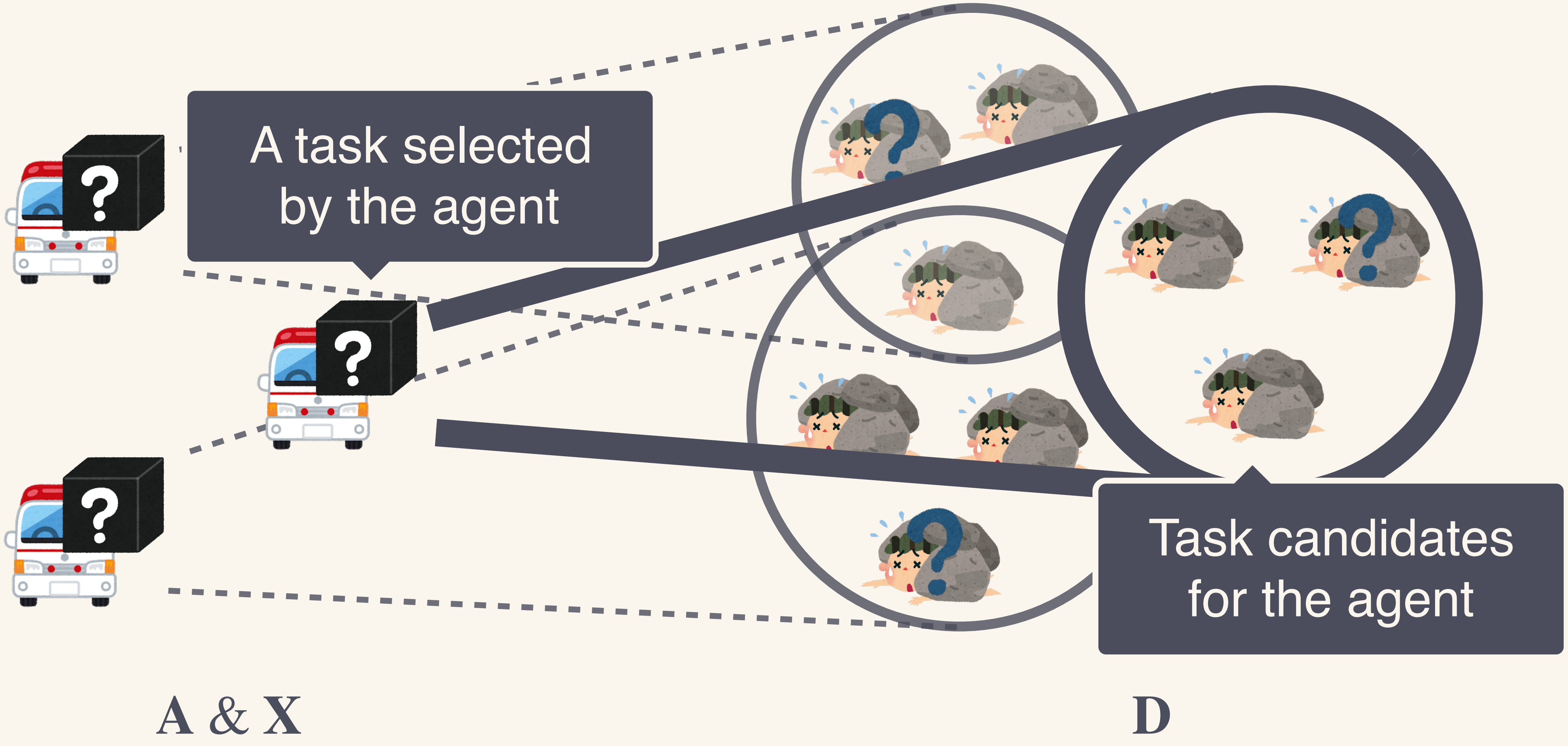
$$\alpha : \mathbf{X} \rightarrow \mathbf{A}$$

Allowing an agent to control 1/more variables; can add attributes & abilities to variables such as location & communication.

※ In most cases, each agent controls 1 variable.

Task Assignment Problem on DCOP

- A task assignment problem can be modeled as a DCOP.



A task selected by the agent

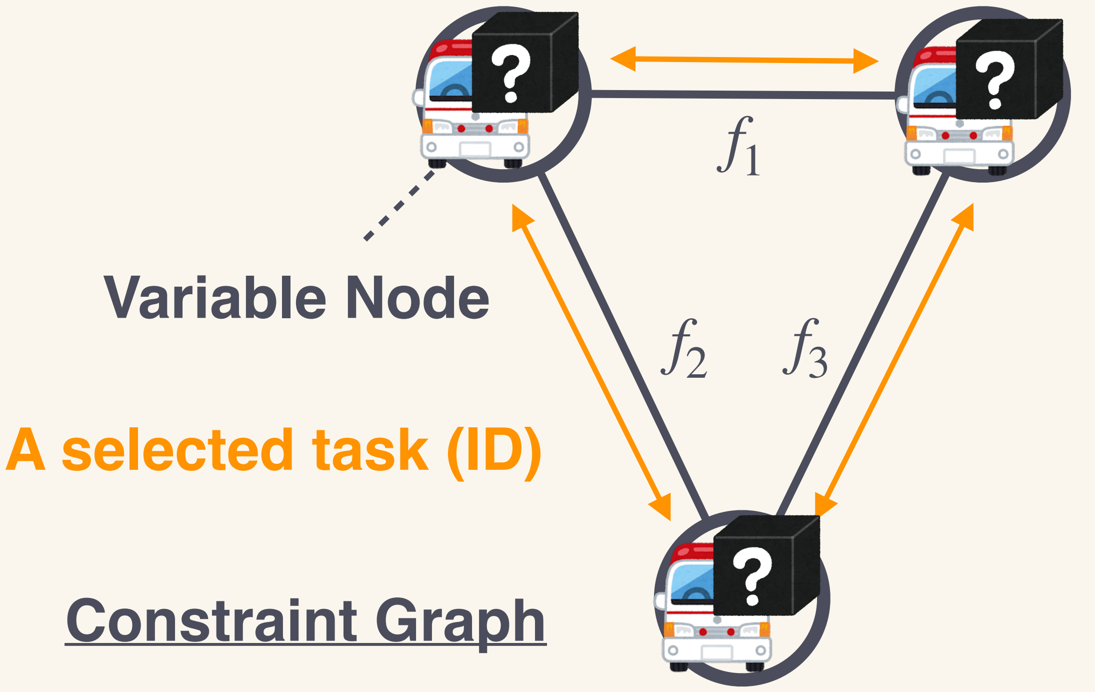
Task candidates for the agent

Two Major Real-Time Algorithms for DCOP

- DCOP algorithms work on a graph that represents a problem.
- Each node selects a value by communicating with the other nodes.

DSA

Uses values of other nodes & random numbers.



f_1

Variable Node

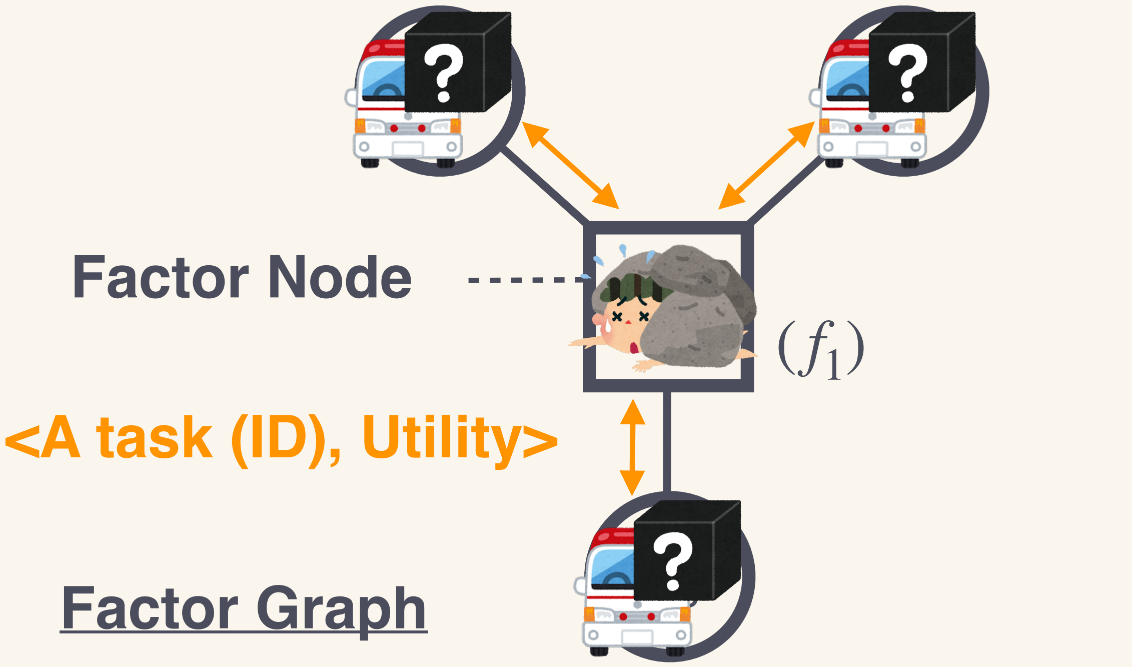
f_2 f_3

A selected task (ID)

Constraint Graph

Max-Sum

Uses value & its utility pairs that were calculated on other nodes.



Factor Node

(f_1)

<A task (ID), Utility>

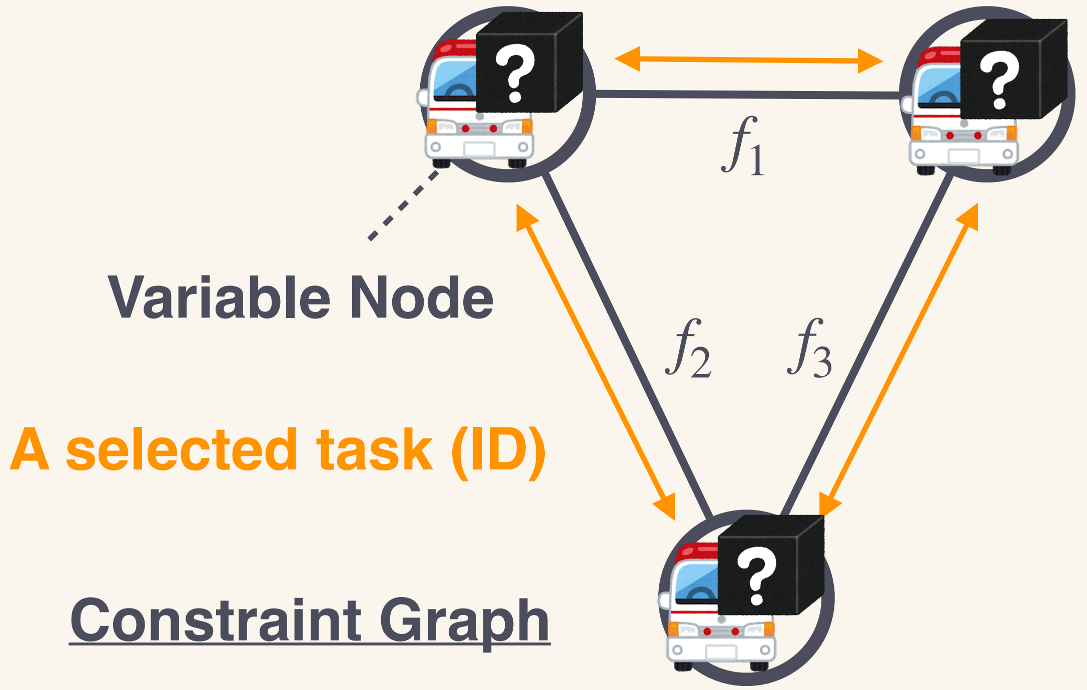
Factor Graph

Two Major Real-Time Algorithms for DCOP

- DCOP algorithms work on a graph that represents a problem.
- Each node selects a value by communicating with the other nodes.

DSA

Uses values of other nodes & random numbers.



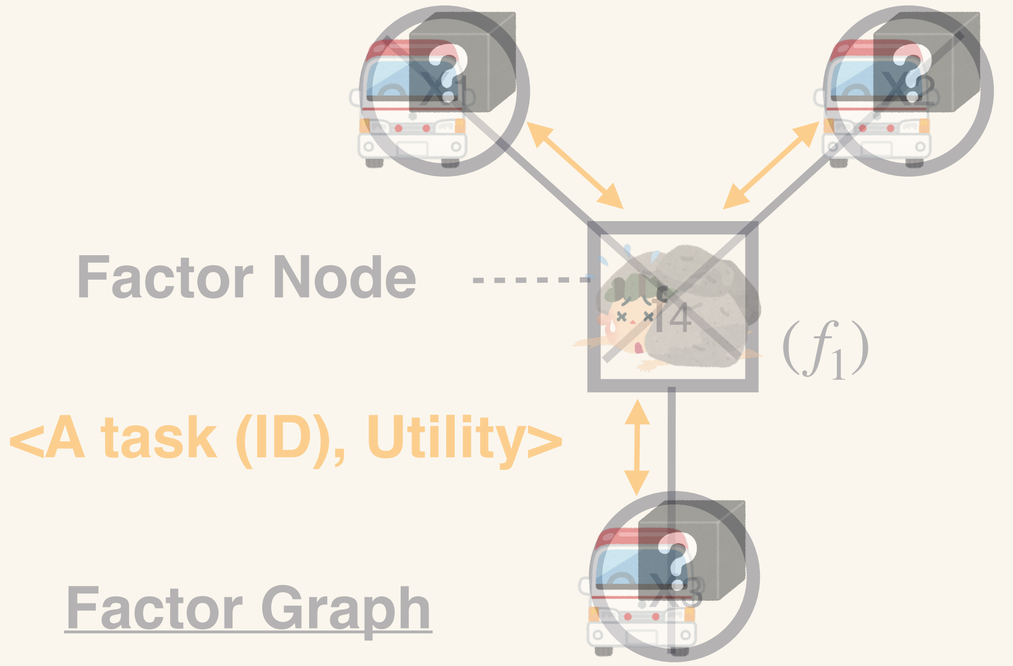
Variable Node

A selected task (ID)

Constraint Graph

Max-Sum

Uses value & its utility pairs that were calculated on other nodes.



Factor Node

<A task (ID), Utility>

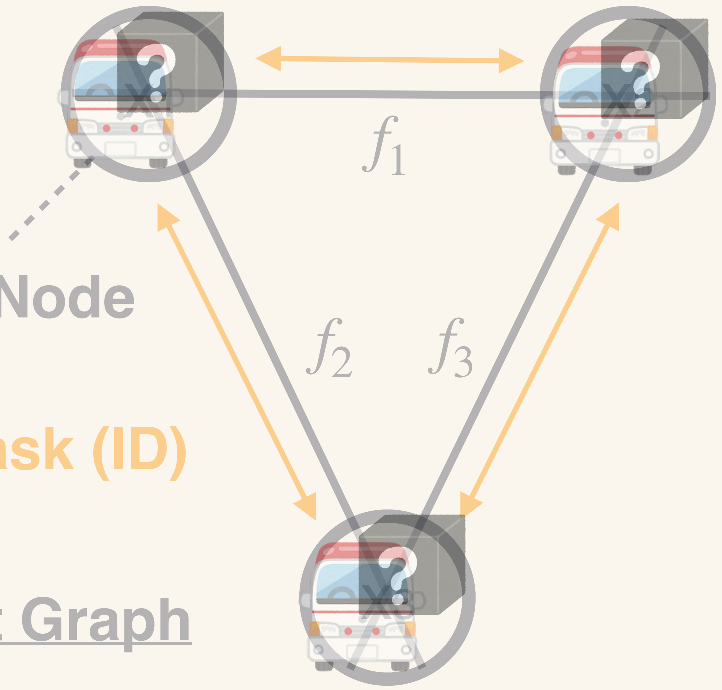
Factor Graph

Two Major Real-Time Algorithms for DCOP

- DCOP algorithms work on a graph that represents a problem.
- Each node selects a value by communicating with the other nodes.

DSA

Uses values of other nodes
& random numbers.



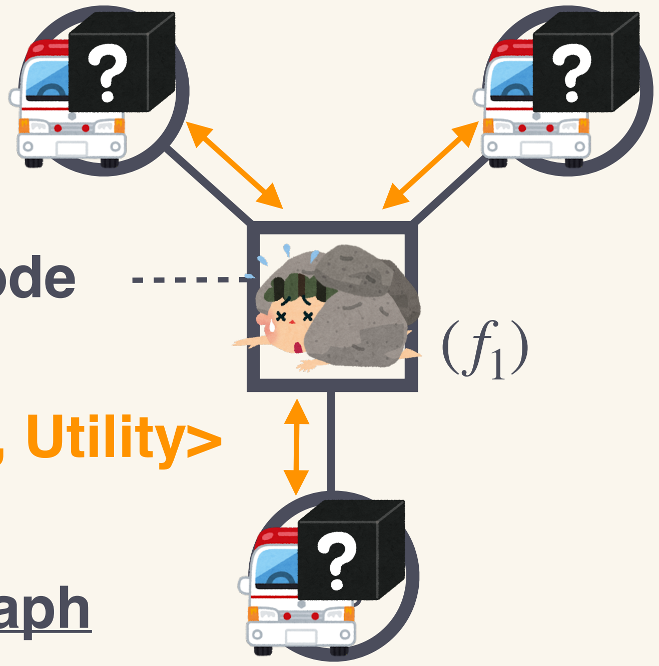
Variable Node

A selected task (ID)

Constraint Graph

Max-Sum

Uses value & its utility pairs
that were calculated on other nodes.



Factor Node

<A task (ID), Utility>

Factor Graph

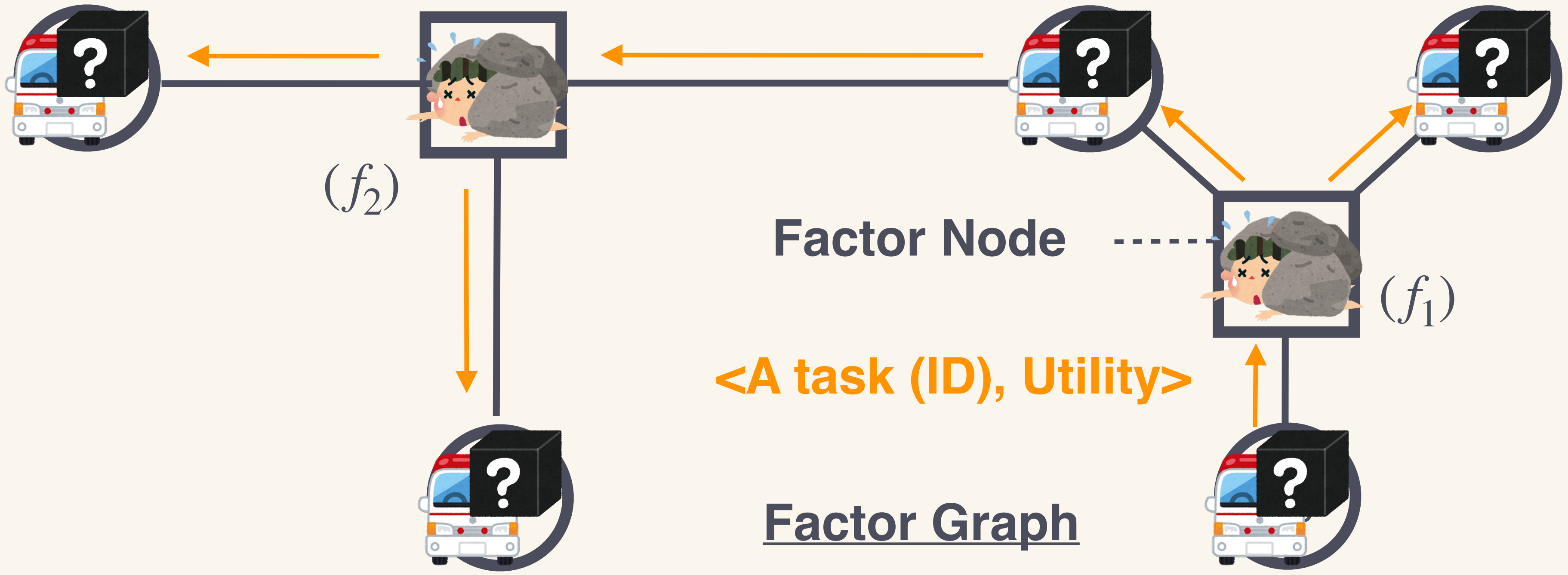
Two Major Real-Time Algorithms for DCOP

- DCOP algorithms work on a graph that represents a problem.
- Each node selects a value by communicating with the other nodes.

Max-Sum

Uses value & its utility pairs that were calculated on other nodes.

The utilities are propagated to the entire graph.



Two Major Real-Time Algorithms for DCOP

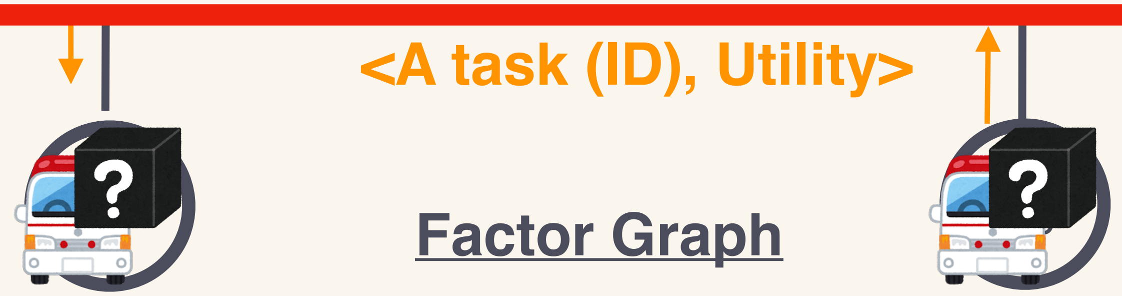
- DCOP algorithms work on a graph that represents a problem.
- Each node selects a value by communicating with the other nodes.

The utilities are propagated to the entire graph.

Max-Sum

Uses value & its utility pairs
that were calculated on other nodes.

DCOP algorithms need to communicate repeatedly.
→ This is difficult for the RRS system.



Related Research: RMASBench

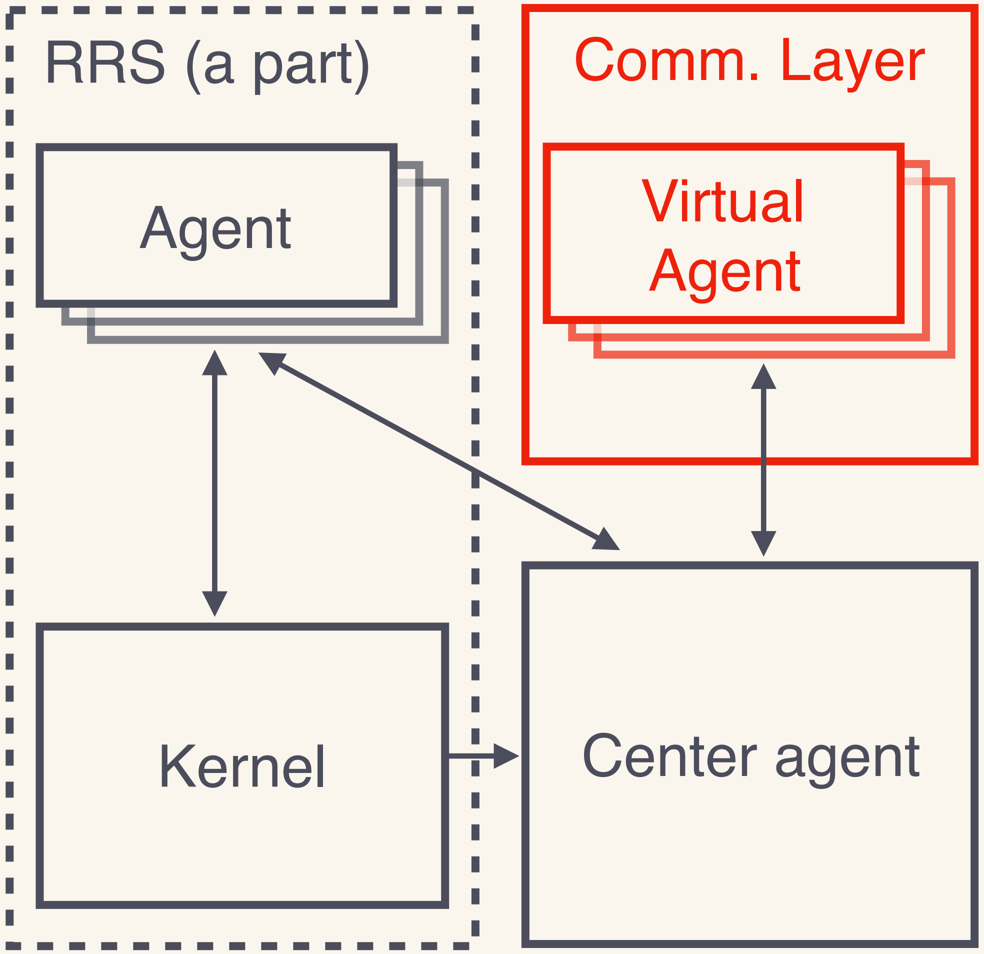
- A benchmark system for DCOP algorithm using the RRS system.
- Its purpose was unclear to our community.
- It is incompatible with ADF.

- It introduced an independent unconstraint communication system on RRS.

[DSA needs 60 communication round trips to solve a problem.

RRS system needs 2 steps for each round trip communication.

→ It is difficult to utilize a DCOP algorithm in 200 - 300 steps.



Purpose

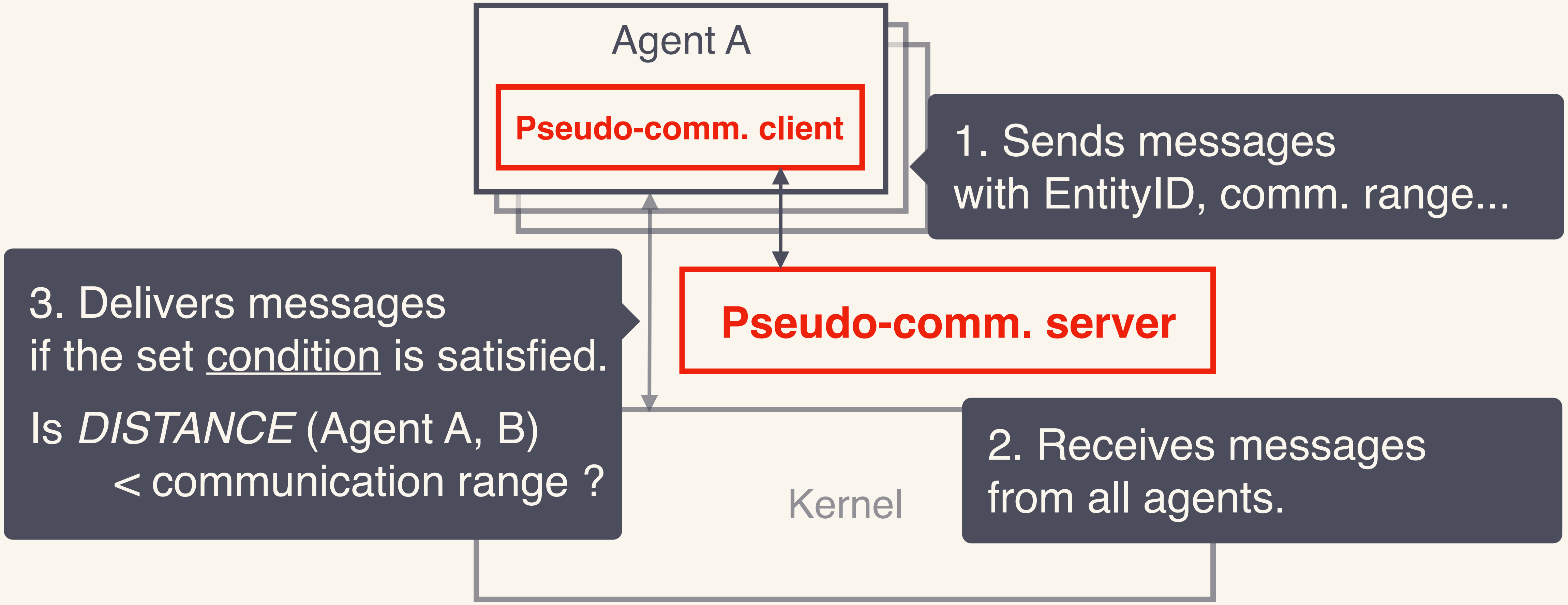
- We propose an environment in which DCOP algorithms can be used for RRS task assignment problems.
  - Our aim for this environment was to create a mechanism that follows the current RRS system as closely as possible.

We can now implement the task assignment module on ADF.
→ We can also use other ADF modules.

However, we really need
a new communication system.

New Communication System

- DCOP algorithms need a communication system that is not restricted by the RRS discrete time.
- We introduce a new communication system using a server & clients.



1. Sends messages with EntityID, comm. range...

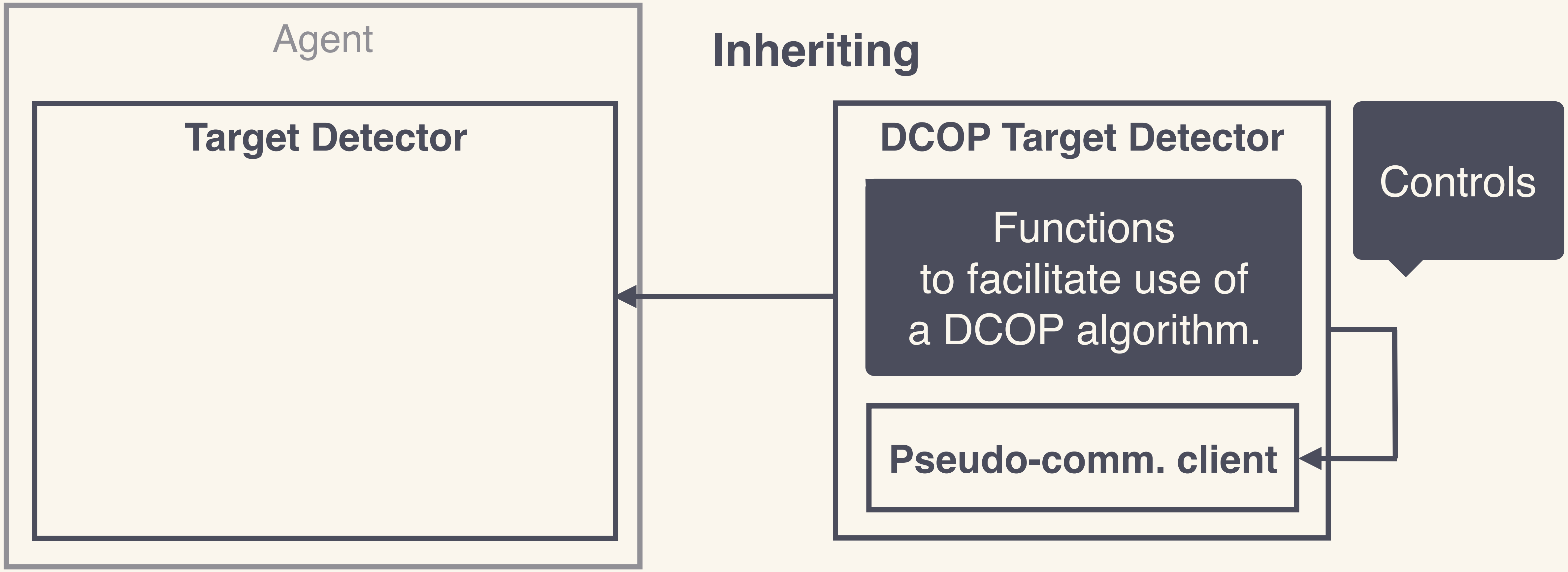
2. Receives messages from all agents.

3. Delivers messages if the set <u>condition</u> is satisfied.

Is *DISTANCE* (Agent A, B) < communication range ?

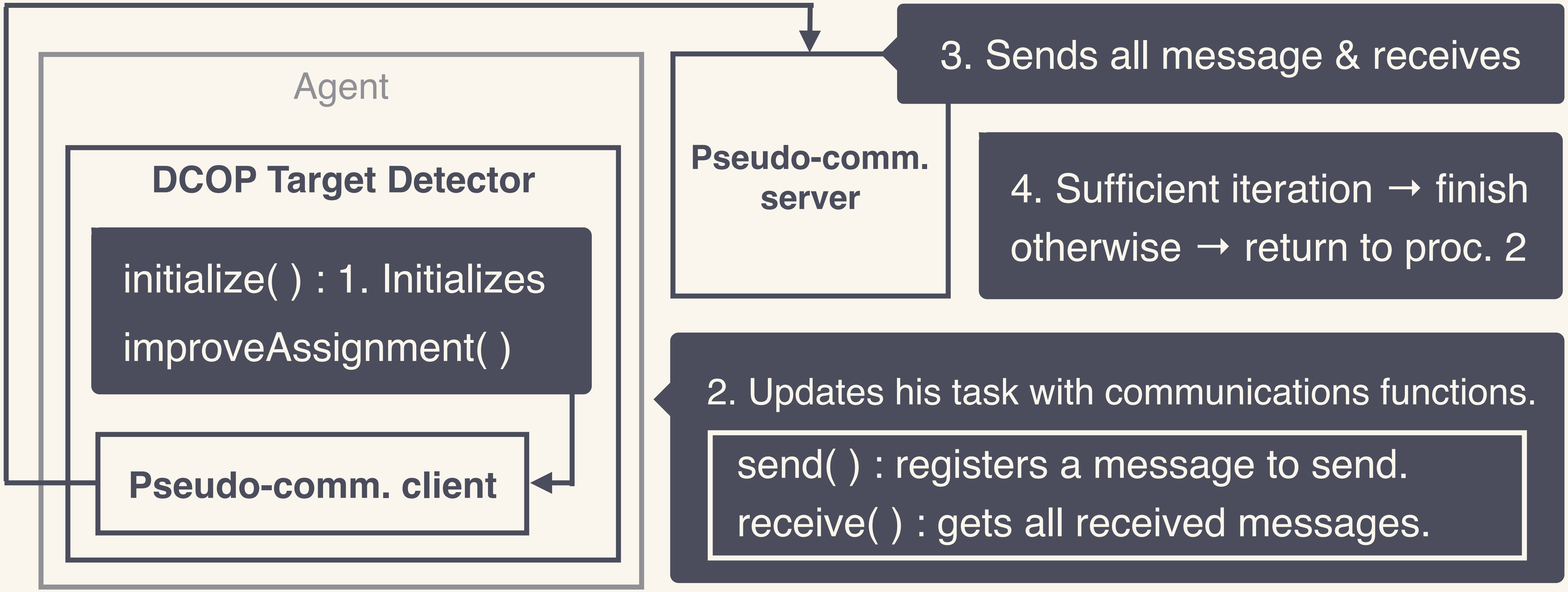
ADF Extension

- We design a task assignment module using the DCOP algorithm & the new communication system.



ADF Extension

- We design a task assignment module using the DCOP algorithm & the new communication system.



initialize( ) : 1. Initializes

improveAssignment( )

2. Updates his task with communications functions.

send( ) : registers a message to send.
receive( ) : gets all received messages.

3. Sends all message & receives

4. Sufficient iteration → finish
otherwise → return to proc. 2

Procedures for running Max-Sum

① Implement a task assignment module using Max-Sum on our environment.

② We also implement Closest & Greedy to make comparisons.

③ Make them run simulations on some conditions (30 times / condition).

Agents : Only Ambulance Teams

Scenarios : VC3 / Eindhoven3 of RoboCup 2018

Agents are diverged / converged

Communication Range : the size of map x 1/4 or 2/4 or 3/4 or 4/4

④ Compare simulation scores.

Implementing Max-Sum (Ambulance Team)

- We describe 3 important points to use Max-Sum.

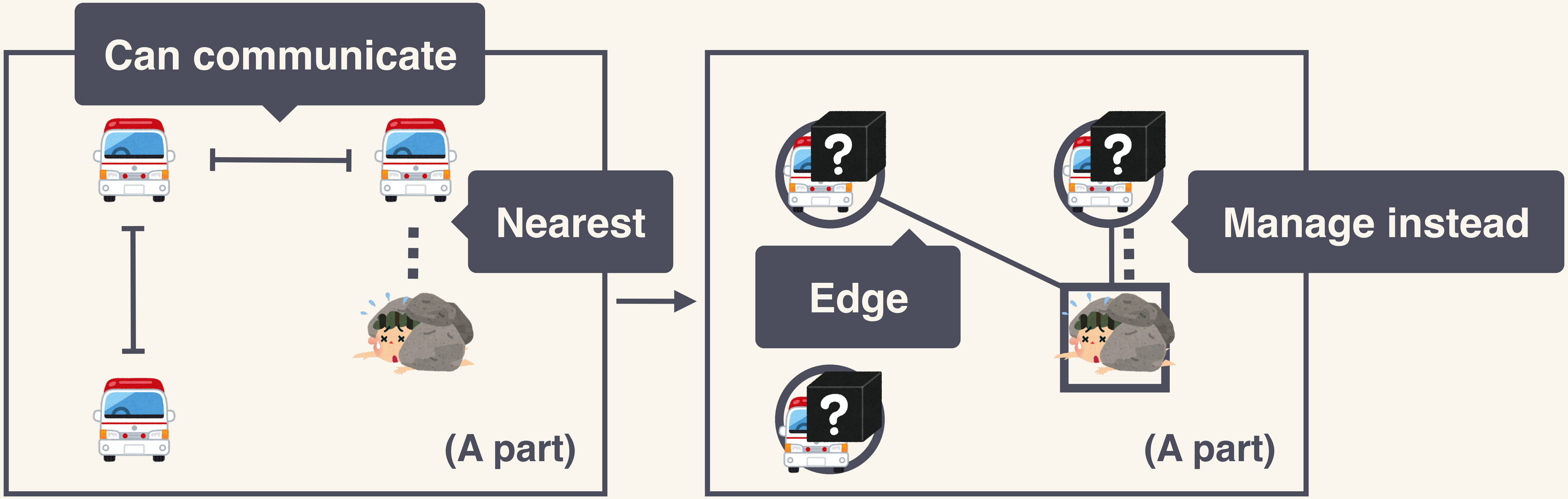
① We had to make the module form a graph based on its environment & run the algorithm on each step to handle a dynamic environment.

② We need to design how to form the factor graph.

③ We also need to design the utility functions.

Forming a Factor Graph

- An agent (variable) is regarded as a variable node, a task related to a utility function is regarded as a factor node.
- Each agent manages own node → 1 graph is created by all agents.
- The nearest agent to each task also manages its factor node.
- An edge created when 2 agents communicate with each other.



Can communicate

Nearest

(A part)

Edge

Manage instead

(A part)

Defining Utility Functions

- $\mathbf{F} = \{ f_1, \ldots, f_k \}$: a set of utility functions.
- The purpose of Ambulance Teams is to rescue as many Civilians as possible.
- We designed the utility functions to minimize the movement costs & assign a sufficient number of agents to each task.

$$
\begin{aligned}
\mathit{Objective\ Function}\ \mathbf{F}_g(\mathbf{X}) &= \sum_{f \in \mathbf{F}} f(\mathbf{X}) \\
&= \sum_{x \in \mathbf{X}} \underline{\mathit{COST\ of\ x}} + \sum_{d \in \cup D_i} \underline{\mathit{PENALTY\ of\ d}}
\end{aligned}
$$

Movement cost

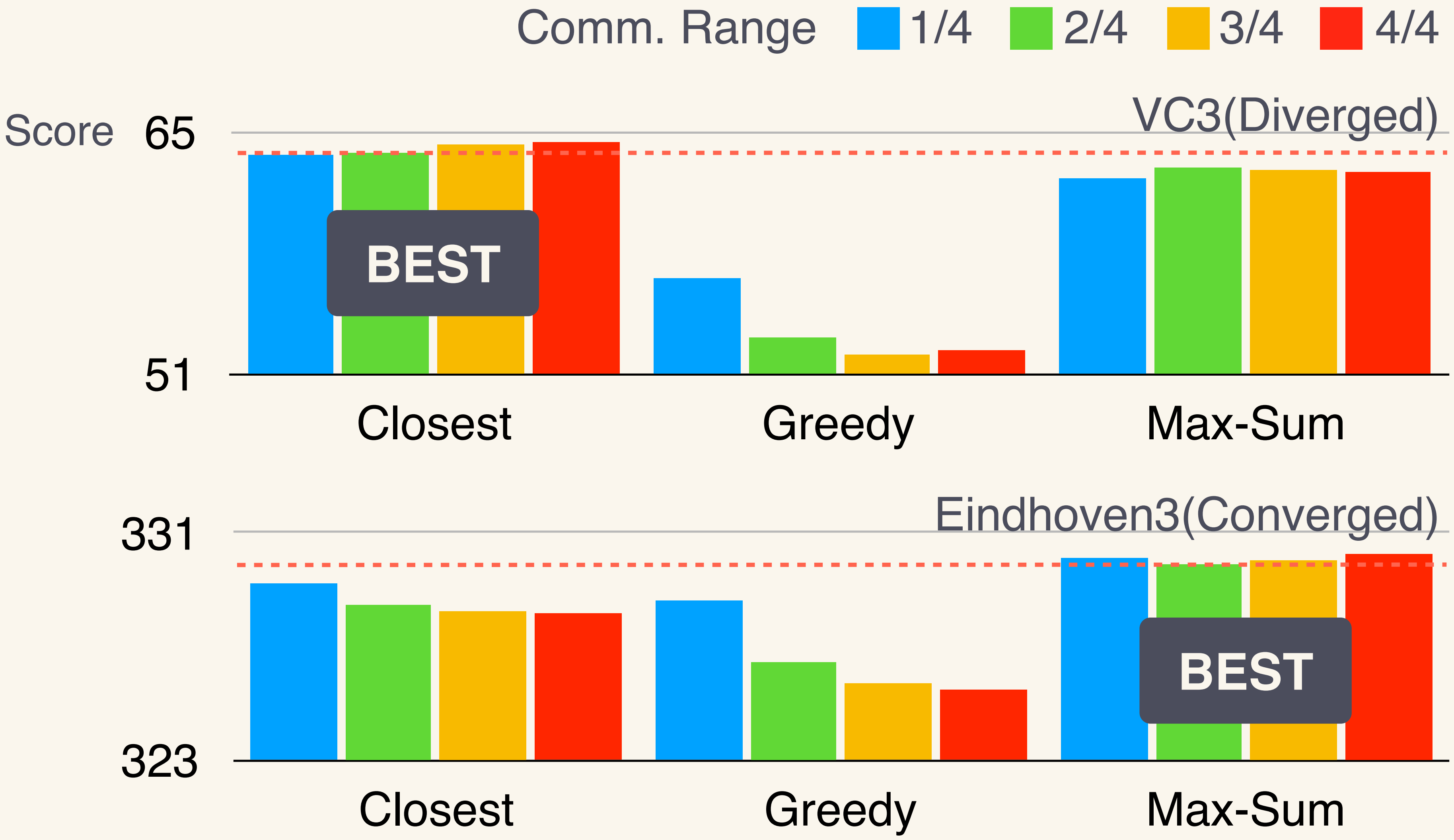
→ used in each variable node

Penalty arising from an insufficient no. of agents

→ used in each factor node

Results

- Max-Sum can result stable & high scores.
- It even works well in scenario where agent have converged.



Conclusion

① DCOP framework is useful for solving task assignment problems.

② We proposed an environment for utilizing DCOP on RRS.

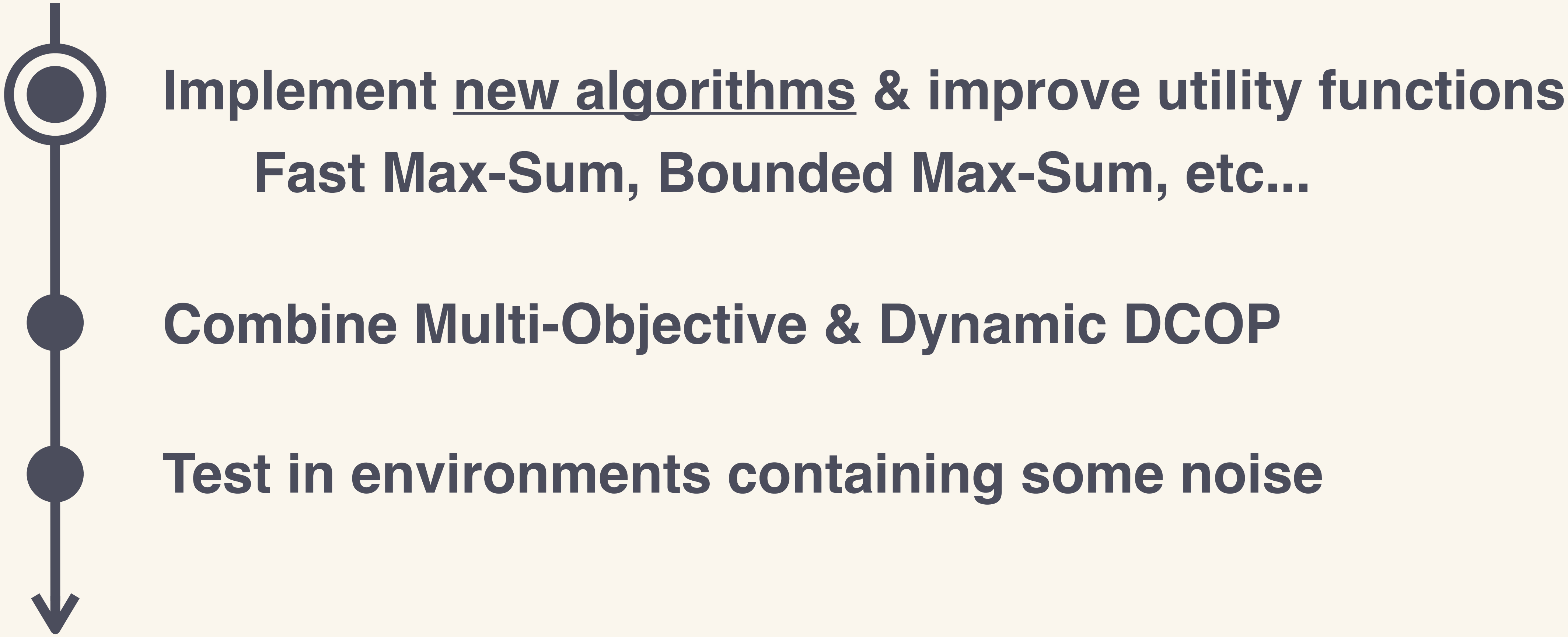
- It introduced a new communication system.
- It provides an ADF task assignment module to use a DCOP algorithm & the communication system.

③ We conducted simulations in our environment & showed their results.

→ Max-Sum could produce high, stable scores on RRS.

Contribution

- We provided an environment that allows DCOP algorithm utilization on RRS.
- We will research DCOP algorithms based on the following themes:



Implement new algorithms & improve utility functions

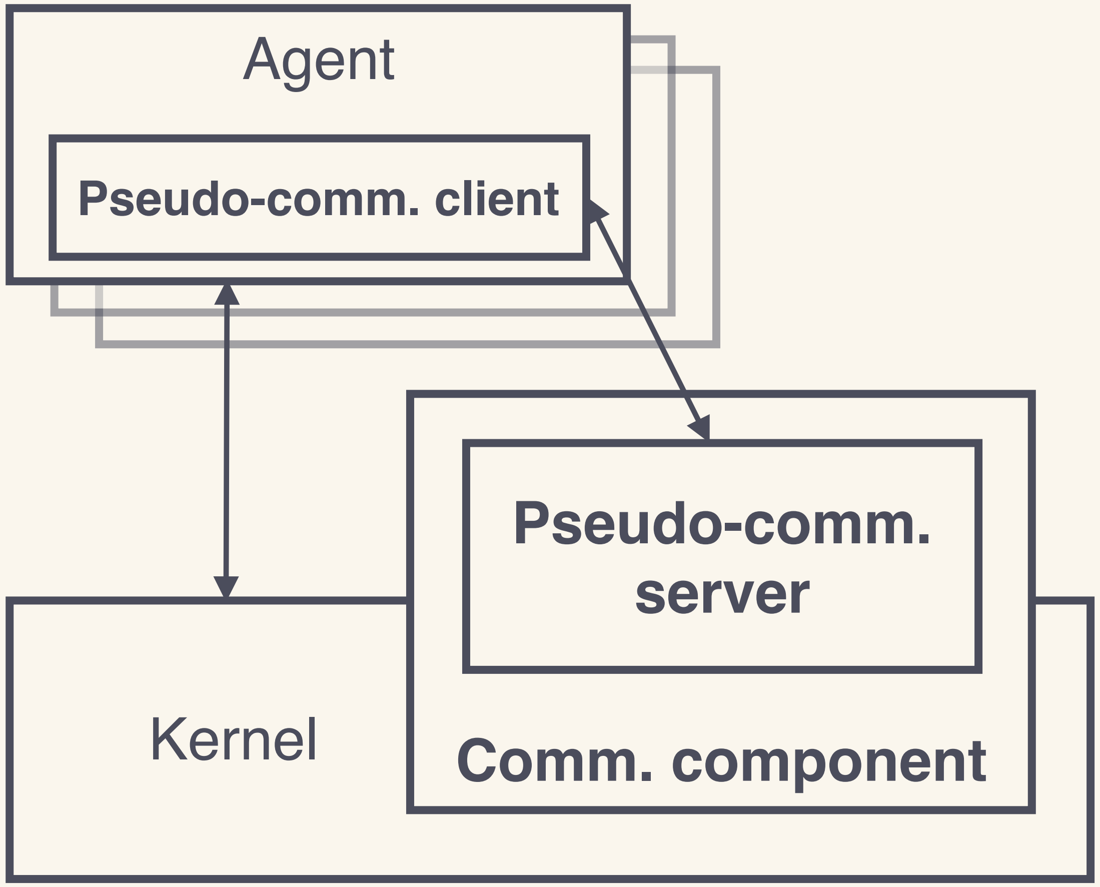
Fast Max-Sum, Bounded Max-Sum, etc...

Combine Multi-Objective & Dynamic DCOP

Test in environments containing some noise

Future Work

- We will attempt to allow the communication component of RRS kernel control the pseudo-communication server & work to make the server easier to configure & use.


Agent
Pseudo-comm. client
Pseudo-comm. server
Kernel
Comm. component