International Journal of Smart Computing and Artificial Intelligence
International Institute of Applied Informatics
2021, Vol. 5, No. 1, 33 – 50

Task Decomposition and Role Sharing for Real-time Human-AI Swarm Collaboration

Sotaro Karakama [*], Natsuki Matsunami [*], Masayuki Ito [*]

Abstract

In spite of the impressive advances in artificial intelligence (AI), close collaboration between humans and AI systems is still difficult to achieve. To overcome this problem, we designed AI agents with a behavior tree that enables us to know what they are trying to do, and by using a consensus building algorithm, that is, a contract net protocol, a human and a group of AI agents were put together as one team. Taking advantage of this architecture, we designed an approach to decomposing cooperative tasks into appropriate roles. The effectiveness and feasibility of this approach were evaluated with teams in a simulated Tail Tag game. Matches were held with up to 29 AI agents and 1 person on one team and 30 people on the other team. The results indicate that our approach works almost evenly with human-human collaboration by sharing roles between a human and AI swarm. By understanding the roles of AI agents, a person can immediately understand the role that he/she should take. For further improvement, we also identified that it is necessary for a person to be able to give concise and global instructions.

Keywords: multi-agent, task decomposition, human-swarm interaction

1 Introduction

The capabilities of artificial intelligence (AI) have greatly advanced, as evidenced by the defeat of the top human players in Go [1], Poker [2] or such video games as Dota 2 [3]. Similarly, attention has recently turned to collaborative teaming of humans and AI in data science [4], game design [5] or for such card games as Hanabi [6] too.

One example of human-AI agent collaboration is *freestyle chess*, in which anyone can play in teams of players and computers. The winners of a 2005 freestyle chess contest were not grandmasters with a state-of-the-art computer but two amateur American chess players using three computers simultaneously. This proved that "*weak human* + *machine* + *better process* was superior to a strong computer alone and, more remarkably, superior to a *strong human* + *machine* + *inferior process*" [7].

AI systems can support humans in making long-term, systematic decisions by utilizing their better memory capability and accuracy. In addition, input data can be processed more quickly to

[*] Mitsubishi Heavy Industries, Ltd., Komaki, Aichi, Japan

assist in tasks that require rapid responses. In contrast, as is known as the "*frame problem,*" [8] current AI systems have difficulty dealing with problems that change the framework of the problems, such as exception handling and the creation of new value. Therefore, it is necessary to clarify the process by which humans understand and use AI systems for high performance on complex reality problems.

Decreasing birthrates and aging populations in many countries will have a major impact in the near future, especially in Japan. In particular, measures addressing labor shortages are urgently required to maintain social infrastructure and security. If people could collaborate with many autonomous machines driven by AI, hereafter *AI agents*, we could overcome and even enhance social welfare even if the number of people is small.

However, attempts to work closely with AI agents in reality, especially in serious and imminent situations such as those related to disaster response or national security, have revealed two problems in particular. First, the effectiveness of these systems is limited by their inability to explain their decisions and actions to a person [9]. Second, since the appropriate degree of support depends on the state of the person, an arrangement mechanism is needed for transferring adequate information between humans and AI agents instantly. As the number of AI agents increases, these problems become more complex. Therefore, most practical autonomous products have been designed to work separately from people [10], to be controlled by a human operator [11], or to perform only explicitly assigned roles that are not shared with a person [12].

In this study, we use an architecture and a prototype of a human-multiple agent interface called "human-swarm interaction" (HSI) [13], which were proposed by *A* et al. (*anonymity for blind review*) [14]. The architecture consists of a behavior tree (BT) [15] and a contract net protocol (CNP) [16] as a mechanism for task sharing between human and AI agents. The use of the BT enables us to know what AI agents are trying to do, while the CNP prioritizes human-AI collaboration without compromising AI agent autonomy.

We confirmed that a human player can cooperate with many AI agents at a close degree, that is, a human team, through an experiment with a simulated game of Tail Tag. In the experiment, we noticed that decomposing tasks into appropriate roles and sharing the information of the roles that AI agents are currently taking helps humans to understand the AI colleagues' intention instantly.

The rest of this paper is organized as follows. First, we introduce related works in the fields of task decomposition and human trust in AI. Then, we discuss the problem domain and the specific environment of the Tail Tag game. In Section 4, we describe the architecture that combines BT and CNP and introduce the implementation of HSI. In Section 5, we explain an example of task decomposition in detail. In Section 6, we demonstrate the effectiveness of task decomposition, limitations, and several remarkable findings as a result. Finally, in Section 7 concludes the paper.

2 Related Works

Building teamwork among autonomous agents for complex, dynamic environments is a major research area of multi-agent domains towards real-world implementations of these agents. In a practical environment, agents must deal with several influences caused by uncertainties such as differing, incomplete, and possibly inconsistent views of their environment. Tambe proposed STEAM as a general model of teamwork that enables a team to act coherently, overcoming such uncertainties of complex, dynamic environments [17]. In STEAM, agents are designed to have an explicit model of teamwork that describes an agent organization hierarchy and the team tasks for the organization. The notion of a role is key in STEAM. A role is an abstract specification of a set of activities an individual or a sub-team undertakes in service of the team's overall activity. Nair et al. also tackled initial role allocation and reallocation upon failures or new tasks [18].

We need to decompose tasks into an appropriate level of granularity before discussing role allocation. Stone et al. introduced a good example of a flexible teamwork structure for RoboCup soccer teams that allows for multi-agent tasks using homogeneous agents to be decomposed into flexible roles [19]. We have noticed that, by using well-defined roles to represent abstracted tasks, coordination is achieved through limited communication and pre-determined procedures as part of a locker-room agreement.

We also need some coordination mechanism to allocate roles among agents. Since we focus on a team consisting of self-interested distributed agents, a distributed approach is required, rather than a purely centralized approach, that can be considered as a single-agent system with many degrees of freedom. A centralized approach may be able to produce optimal plans if the complexity of the problem is sufficiently small. However, there are several critical problems with a purely centralized coordinated system. For instance, communication between the leader and others is critical since the leader determines everything. The point of vulnerability for such tightly coupled teams is the leader and its communication. Moreover, as the number of agents increases, the communication cost also increases.

Market-based coordination is one distributed approach, and it has been proven in the field of multi-robot coordination [20]. Originally, Smith introduced the concept of using an economic model to distributed multi-agent systems as a CNP [16]. The CNP is the most fundamental model that applies economic activities to multi-agent systems.

These related works indicate that smart agents can deal with a practical problem if we can assemble the appropriate solutions to each problem. Hence, the complexity of multi-agent-team decision problems in partially observable environments can be relaxed with communication [21], and it becomes more realistic to make a practical team of agents by decomposing tasks into roles at an adequate granularity and to reduce communication costs with a coordination mechanism. However, none of these works focus on cooperation between human and AI agents. A specific problem arises when human-AI collaboration is needed. A wide range of research has proposed ways to explain AI systems, and some have found that too much explanation can create confusion and degrade trust, and thus, humans prefer simple explanations [22]. In particular, despite the rich capabilities of recent machine learning, the growing complexity of these algorithms has made them difficult to explain to humans. Because we consider developing human-AI collaboration and trust to be the most important goal of our research, we use the old-fashioned multi-agent and multi-robot approach described above.

3 Problem Setting

3.1 Problem Domain

In this study, we aim at enhancing team capabilities in situations where there are serious and imminent problems such as in disaster response, search and rescue, and security threats. Such situations cannot be handled by robots alone since they might threaten people's lives, so humans must consider legal and ethical responsibilities in a timely manner when working with these robots. Furthermore, technical knowledge that can handle these serious problems are also useful for easier situations of human-AI collaboration such as those involving resource mapping and robot sheepdogs.

Moreover, to evaluate human-AI agent collaboration, we decided to compare it with collaboration between people. First, a team consisting of a human and AI agents competes against a team consisting of humans. Next, we confirm the performance of collaboration between the human and AI agents through the results of the matches.

The key features of this problem domain are as follows.

1) multi-agent,
2) team-style symmetry,
3) simultaneous games,
4) games with imperfect information, and
5) the availability of multiple strategies with no persistent equilibria.

We could not find a good abstracted existing environment that satisfies all the features above in prior studies such like RoboCup soccer [23], StarCraft [24], or capture the flag [25]. To focus on the fundamental aspects of the problem, we used a simpler environment and rules than these environments.

As a result, we chose the simple team game Tail Tag, which satisfies all the features above, as the environment for this research, and we constructed a simulation. Tail Tag is a children's game. The players are divided into two teams, and they tuck a piece of ribbon or string into the back of their shorts as a *tail*. Then, they run around the game area trying to catch the tails of the opponent team while defending their own tail. To win this game, it is important to understand the current situation and to quickly estimate the opponent players' tactics. Furthermore, the players need to be flexible in their strategy and cooperate with their teammates, such as by coordinating their actions ("trap him/her in a pincer!") and giving warnings ("look out behind you, run!").

3.2 Tail Tag Game

The environment and the rules of our Tail Tag game are described in Figure 1. A round game area is used to avoid characteristic shapes such as corners. The game features two teams of agents. Team A consists of one person and a group of AI agents, while Team B consists of people. The human player on Team A directly controls his/her agent's movement and communicates with the AI agents through HSI, which we describe later. The AI agents can not only act autonomously but also interact with the human player. The human players on Team B can communicate with their teammates by voice chat while they directly control their own agent. Each human player on either team uses a gamepad with two joysticks to control their agent. The translational velocity is controlled by moving one stick that has two analogue axes (X and Y), and the rotational velocity is controlled by moving the other stick, which has one analogue axis (Z). These settings satisfy key features 1, 2, and 3 in Problem Domain.

An agent catches an agent on the other team when his/her front collides with the back of the other agent of the opponent team. Caught agents as well as agents that leave the game area are eliminated from the game. The winner of this game is the team that either eliminates all adversary agents or has more remaining agents after a time limit of 10 minutes. If two agents on opposing teams simply collide without a capture resulting, they are frozen for a certain length of time after being knocked back. In addition, if teammates collide, they are frozen for the same length of time regardless of their collision pattern. As shown in Figure 1, the field of view of each agent is limited to a circular region. The positions and directions of the friend agents and of the observed adversary agents are shared in an overhead view with the teammates. However, there is large unobserved area outside the fields of view of the teammates. In other words, key feature 4 in Problem Domain is satisfied.

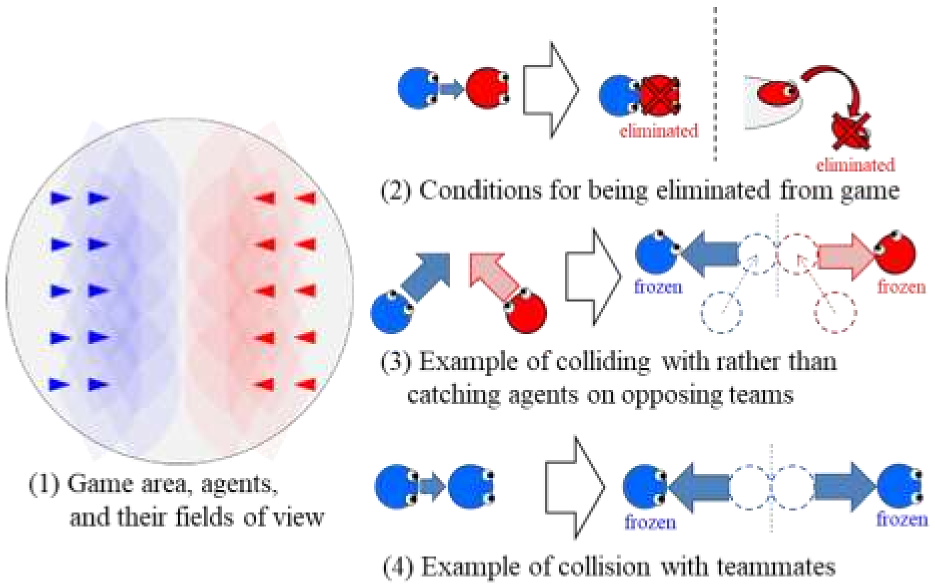

eliminated
eliminated
(2) Conditions for being eliminated from game
frozen
frozen
(3) Example of colliding with rather than catching agents on opposing teams
(1) Game area, agents, and their fields of view
frozen
frozen
(4) Example of collision with teammates

Figure 1: Rules and environment of Tail Tag game used

Success in this game requires collaboration among teammates. For example, since each agent performs the same, it is impossible for an agent to catch a target by using an optimal strategy against the other agent in a one-to-one situation. However, if two teammate agents cooperate, one can intentionally collide head-on with an opposing team agent, causing it to become frozen, which will enable the other teammate to catch it. When considering such cooperation, there can be complex combinations of teammates to cooperate with and targets to share. This is equivalent to key feature 5 in Problem Domain.

4 Overview of Architecture and Interface

4.1 Architecture

As described in Section 3, success in the Tail Tag game requires teamwork. One of the keys to effective collaboration is the ability to update adaptive strategies dynamically to achieve the team's goal while monitoring the situation including the opponent's activities. Since we assume that the AI agents are autonomous, they also have individual strategies based on their local situation. For example, even if the team's goal is to reach a specific area at a certain time, any agents in imminent danger shall give priority to avoiding it. Any agents that are not in imminent danger need to move toward the target location to achieve the team's goal. A centralized approach in which one person directs a group of AI agents would become inefficient as the number of agents increases. Given this scalability problem and the uneven distribution of information, it becomes more attractive to use an architecture in which the individual local strategies of the agents and the team's global strategies are aligned in a decentralized manner.

The architecture consisting of CNP and BT achieves strategic negotiation and alignment.

4.1.1 Contract Net Protocol

CNP [16] is a protocol for decomposing a complex mission into independent tasks and assigning them to multiple agents, analogous to contract bidding in business transactions.

With CNP, an agent (the *manager*) first broadcasts a task-announcement message. Next, any agent capable of performing the announced task sends a bid message to the manager for the task. Then, the manager evaluates the bids received, selects appropriate agents, and sends an award message to the selected agents. These agents selected by the manager are called the *contractor*. Finally, the manager receives a report message from the contractors detailing the results of the execution. Thus, CNP forms a hierarchical task allocation structure by top-down processing.

The main feature of CNP is mutual selection. In other words, when entering into a contract, managers and contractors have independent utility functions for bidding and awarding. Thus, CNP is a protocol that enables negotiation between multiple agents regarding task assignment.

4.1.2 Behavior Tree

The BT is a plan representation and decision-making tool used to model and control autonomous agents. It was created in the game industry [15] and has been widely adopted by the game development community. It is commonly used to model non-playable characters and is viewed as an alternative to finite-state machines and hand-coded rules via scripting.

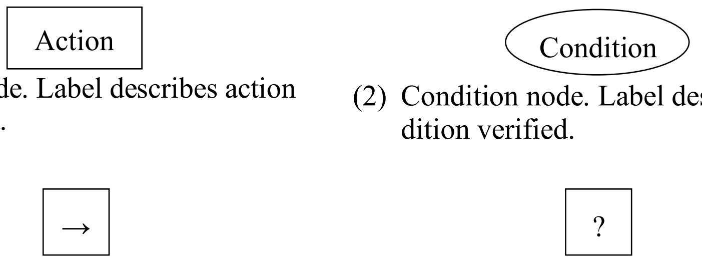
BTs benefits for real-time, constrained, and complex applications are as follows.

- Ease of understanding and control,
- Fast run-time execution,
- Quick implementation using graphical tools, and
- Flexibility and scalability; their complexity can be scaled up and down as needed, and subtrees can be easily reused.

These benefits help human players to understand the intent and purpose of agent behavior in human-AI collaborative systems. The typical nodes in the BT are action, condition, sequence, and selector. Figure 2 shows a graphical representation of these nodes.

As shown in Figure 3, we defined two action domains: cooperative tasks and individual tasks. Cooperative tasks are driven by tasks contracted to an agent by the CNP, and individual tasks are tasks executed at the discretion of the agent. There are four types of individual tasks. *Leave the edge* [corresponding to individual task (1) in Figure 3] and *collision avoidance* [corresponding to individual task (2) in Figure 3] are essential tasks for safety. *Track and catch the target* [corresponding to individual task (3) in Figure 3] follows an agent's tactics and requires urgent action based on local information known to the agent. It is important to note that *track and catch the target* is the simple behavior of going around behind the target closest to oneself and not cooperating with teammates. Finally, the search task for finding enemies requires a relatively long-term strategy to be achieved, and humans are usually better than AI agents at such tasks under uncertainty. Therefore, *random search* [corresponding to individual task (4) in Figure 3] by an AI agent is designed to be simple behavior that involves moving around randomly.

A BT basically evaluates a graph from left to right, so *CNP tasks* that require teammates to cooperate are designed to be to the left of individual tasks.



(1) Action node. Label describes action performed.

(2) Condition node. Label describes condition verified.

(3) Sequence node: set of behaviors, all of which must be completed for sequence to succeed.

(4) Selector node: set of behaviors, any one of which must be completed for selector to succeed.

Figure 2: Graphical representation of action (1), condition (2), sequence (3), and selector (4) nodes.

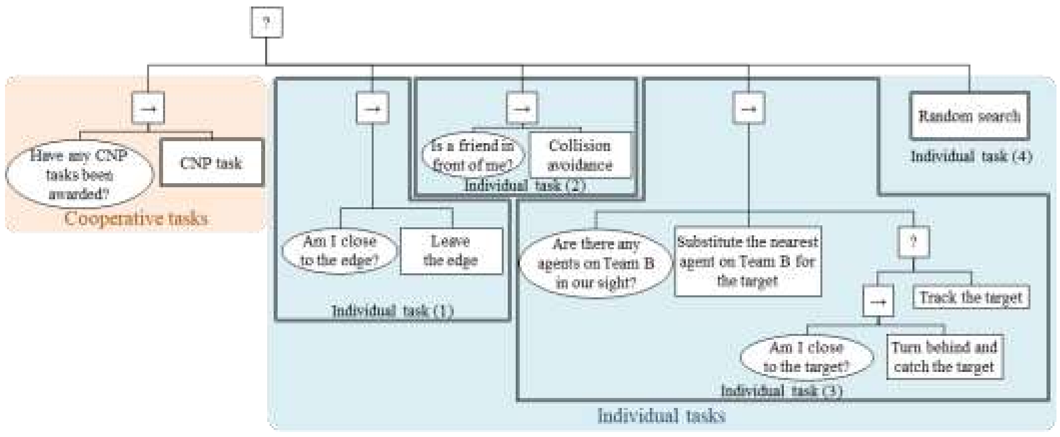

?
→
Have any CNP tasks been awarded?
CNP task
Cooperative tasks
→
Am I close to the edge?
Leave the edge
Individual task (1)
→
Is a friend in front of me?
Collision avoidance
Individual task (2)
→
Are there any agents on Team B in our sight?
Substitute the nearest agent on Team B for the target
?
→
Track the target
Am I close to the target?
Turn behind and catch the target
Individual task (3)
Random search
Individual task (4)
Individual tasks

Figure 3: Behavior tree of AI agent on Team A

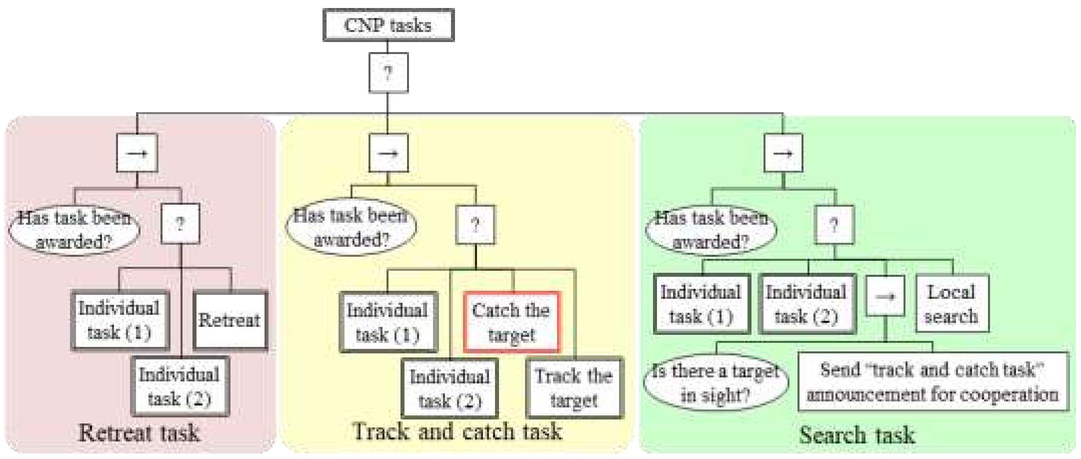


Figure 4: Subtree of CNP tasks. Individual tasks (1) and (2) correspond to Figure 3. Red box represents *catch task* that is decomposed in this paper.

4.1.3 Prioritization of CNP Tasks

As we described in Subsection 4.1, the architecture must align the needs of cooperative tasks and individual tasks. This is done by making the priority of all tasks flexible. As shown in Figure 4, the *search task* and *retreat task* are the same in terms of directing the agents to move towards the designated position, but their urgencies differ. In particular, continuing the *search task* is not appropriate for an agent that is likely to catch an opponent agent. In this situation, the AI agent should switch from the *search task* to the *track and catch tasks*. In contrast, an agent should not switch from the *retreat task* to the *track and catch tasks*, while it would be acceptable to switch to the *leave the edge* or *collision avoidance tasks* of the individual tasks for safety reasons. We thus prioritized the tasks as follows.

- Prioritize tasks that ensure safety. Therefore, the *retreat task* has higher priority than the *track and catch* and *search tasks*. Similarly, *leave the edge* and *collision avoidance* have higher priority than the other tasks.
- An urgent task based on local needs has higher priority than non-urgent tasks based on global needs. For example, *track and catch tasks* have higher priority than the *search task.*

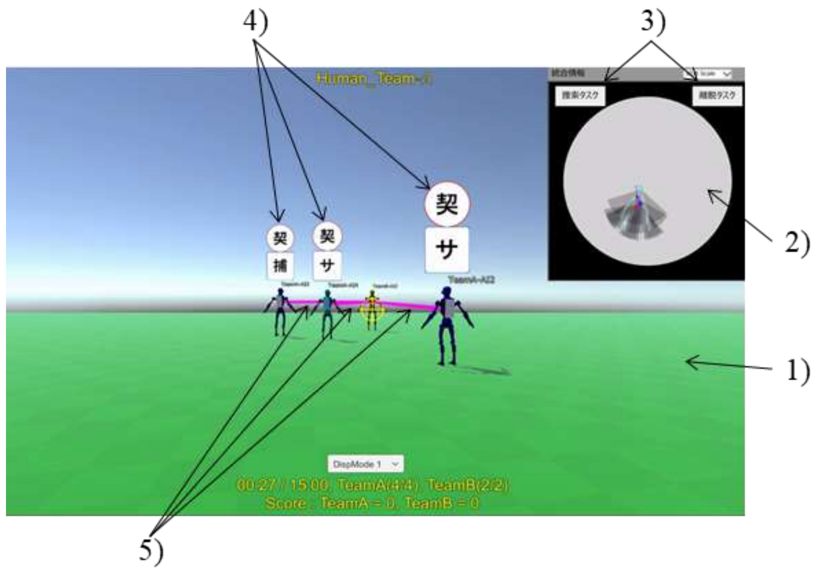
As we described, we assume that people are better at making decisions on the basis of uncertain information, and AI agents are better at performing actions on the basis of local information. For this reason, we designed the architecture so that the *search* and *retreat tasks* are announced by the human player, and the AI agents bid for them.

Regarding the *track and catch tasks*, it is necessary to make a decision on the basis of both uncertain global information and local information. Therefore, these tasks are flexibly designed to be announced by both of human player and AI agents. In this paper, the most important *catch task* (corresponding to the red box in Figure 4) is decomposed into roles. With this task decomposition, a comparison experiment is carried out from the viewpoint of whether a human can flexibly cooperate while understanding the intention of AI agents. The approach to task decomposition is discussed in detail in Section 5.

4.2 Human-Swarm Interaction

In this subsection, we introduce our prototype interface for HSI, shown in Figure 5. The players on Team A use a game controller or keyboard to control their agent and use a CNP interface to interact with the AI agents. There are two types of CNP interface. For the first, the human player on Team A can announce a *search task* and *retreat task* by using a mouse and overhead view. For the second, the human player can announce *track and catch tasks* and change the target as needed by using buttons assigned to the game pad. In addition, the intentions of Team A's AI agents are represented by task/role symbols and a targeting line. The symbols correspond to each BT task/role, and the targeting line represents the target point.

Team B had almost the same interface screens as Team A. The main difference from Team A's interface is that the Team-B members could communicate with each other through voice chat by using a headset. Table I compares the interfaces.


4)
3)
Human_Team-A
契
サ
契
捕
契
サ
TeamA-AI2
2)
1)
DispMode 1
00:27 / 15:00, TeamA(4/4), TeamB(2/2)
Score : TeamA = 0, TeamB = 0
5)

1) First person view camera, 2) overhead view, 3) CNP interface,
4) task/role symbols and 5) targeting line

Figure 5: Display layout of human player on Team A

Table 1: Interface comparison

User / *Interface*		*Team-A players*	*Team-B players*
Display function	1st person	Available	Available
	Overhead	Available	Available
	CNP interface	Available	Unnecessary
	Task/role symbols	Available in some experimental trials	Unnecessary
	Targeting line	Available in some experimental trials	Unnecessary
Game controller		Available	Available
Keyboard		Available	Available
Mouse		Available	Unnecessary
Headset		Unnecessary	Available

5 Task Decomposition of Catch Task

5.1 Overview

This section describes the approach to task decomposition as a condition for comparison in the experiments.

As we described in Section 3, in our Tail Tag game, an agent cannot catch the target without cooperating with another agent when agents of both sides take optimal strategies against each other in a one-on-one situation because the agents perform the same in terms of maneuverability. Therefore, some number of agents need to form a temporary team to catch a target. At least two agents are required to catch an opponent, and each has different roles: confine and catch. To catch the target, an agent has to collide with the target from the front or side to *confine* and force the target to be in a stuck (frozen) state for a while; then, the other teammate has to rush into the target's back. However, a target that a human is playing against can easily notice that he/she is being targeted by these two opponent agents and escape from being confined. Moreover, since the agent that took the role of *confine* in the *catch task* must be vulnerable from being stuck for a while, to protect its back from another opponent agents, we need another form of support for the agent of the *confine role*. Therefore, we added the role of support to the *catch task*. As a consequence, we decomposed the *catch task* into three roles: *support*, *confine*, and *catch*. The basic unit of a temporary team for this task is a three-man cell.

In this study, we compared two cases. When teammates agree to form a team to perform the *catch task*, one case is where the three roles of the task are changed dynamically, and the other is where only one role is executed and not changed. In the latter case, agents take only the *catch role*. In the following subsections, we give details on each role and the way of changing roles.

5.2 Calculation of Time to Arrive at and Position to Reach Target

Estimating the position to reach the target and the time to arrive at that position is important for the *catch task*. For example, suppose an agent freezes a target in the hope that a teammate will catch it. However, if the teammate cannot arrive at the target while the target is frozen, confining the target is clearly futile.

The time to arrive at the target and the position to reach the target are calculated under the assumption that the movement direction and velocity of the target will not change in the near future. The position of the *catch and track task* agent is denoted as $\boldsymbol{P_B}$, and the maximum velocity of the catching agent is denoted as V_B. When the target agent's position is set to $\boldsymbol{P_R}$ and the target's velocity vector is set to $\boldsymbol{V_R}$, the time to arrive T_a is obtained by equation (1).

$$T_a = \frac{-\boldsymbol{R}\cdot\boldsymbol{V_R} - \sqrt{(\boldsymbol{R}\cdot\boldsymbol{V_R})^2 - |\boldsymbol{R}|^2\left(|\boldsymbol{V_R}|^2 - {V_B}^2\right)}}{|\boldsymbol{V_R}|^2 - {V_B}^2} \tag{1}$$

, where $\boldsymbol{R}$ is a relative position vector from the capturing agent to the target: $\boldsymbol{R} = \boldsymbol{P_R} - \boldsymbol{P_B}$.

If there is a real number solution for the time to arrive T_a, the position to reach $\boldsymbol{P_r}$ is denoted as equation (2).

$$\boldsymbol{P_r} = \boldsymbol{P_R} + T_a\boldsymbol{V_R} \tag{2}$$

Here, if the position to reach is outside the game area, the time to arrive T_a and the position to reach $\boldsymbol{P_r}$ are treated as having no solution.

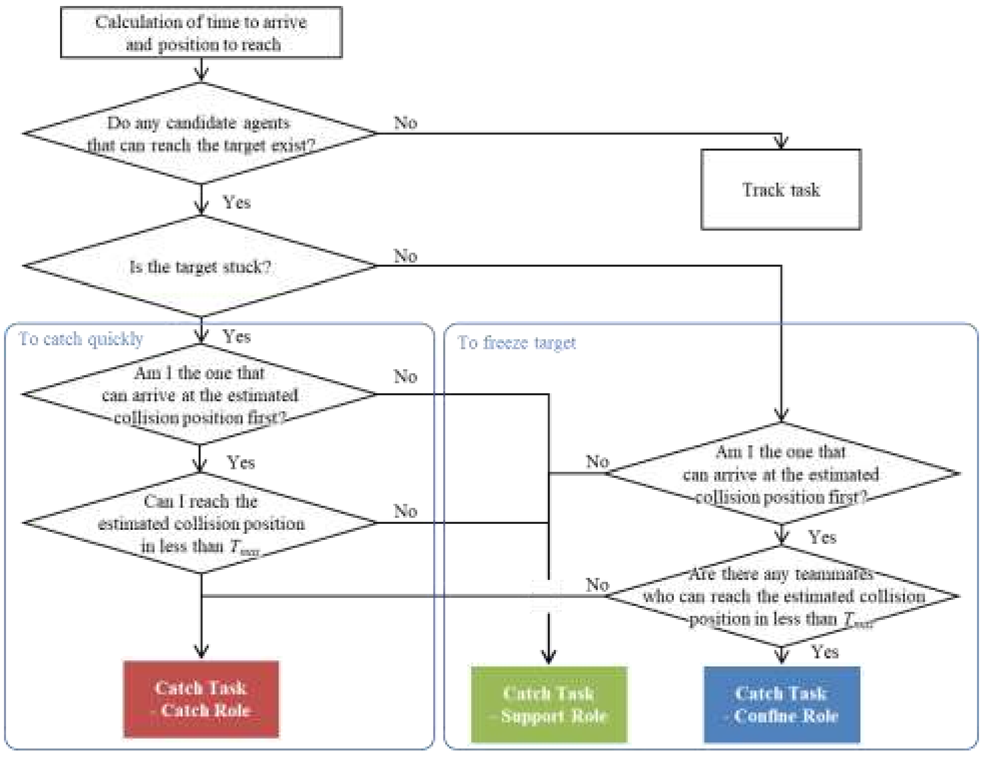

Calculation of time to arrive and position to reach
Do any candidate agents that can reach the target exist?
No
Track task
Yes
Is the target stuck?
No
To catch quickly
Yes
Am I the one that can arrive at the estimated collision position first?
No
Yes
Can I reach the estimated collision position in less than T_{max}
No
To freeze target
No
Am I the one that can arrive at the estimated collision position first?
Yes
No
Are there any teammates who can reach the estimated collision position in less than T_{max}
Yes
Catch Task - Catch Role
Catch Task - Support Role
Catch Task - Confine Role

Figure 6: Flowchart for assigning roles of catch task

5.3 Roles of Catch Task

As shown in Figure 6, we design the execution conditions of each role with reference to the cooperation between humans when our Tail Tag game is played. In other words, human-AI agent collaboration is designed to imitate interaction between people and to determine the roles of human and AI agents. We use the calculation of the time to arrive at and the position to reach the target to determine the situations where each role should be executed.

First, if the target runs away in the opposite direction to all cooperative agents, these agents do not have a solution from calculating the time to arrive, so we then assign the *track task* to them as an exception. In the *track task*, the agent moves to the current position of the target agent at the maximum velocity.

Next, the state of the target agent is evaluated to determine whether it is frozen or not. If the target agent is frozen, the cooperative agents try to catch it immediately. When the agent is one that can arrive at the estimated collision position first among cooperative agents and the time to arrive is smaller than the maximum allowable time T_{max}, the *catch role* will be assigned. The maximum allowable time T_{max} is a parameter, and we set T_{max} shorter than the agent freezing time. Conversely, if the target agent is not frozen, the cooperative agents try to confine the target so as to stop it. When the agent is one that can arrive at the estimated collision position first among cooperative agents and the time to arrive of the other cooperative agents is smaller than the T_{max}, the *confine role* is assigned. In other words, the agent assumes that the teammates will catch the target after he/she has successfully confined the target. If the time to arrive of the other cooperative agents is larger than T_{max}, the *catch role* is assigned. This means that an agent attempts to catch the target without cooperation since there are no teammates that can form a temporary team in a timely manner. Finally, a *support role* is assigned to the agent that does not have a *catch role* or *confine role*.

As a result, each role of the *catch task* can be interpreted as follows.

- *Catch Role*: The role of capturing the target. This agent is expected to move towards and behind the target and touch its back. If the agent is faced with the back of the target agent, it moves to the current target position and catches it.
- *Confine Role*: The role of confining the target agent to stop it. This agent moves to the current position of the target agent regardless of the target's direction.
- *Support Role*: The role of supporting the other cooperative agents. In this study, we implemented this role to have the same behavior as the *confine role*.

As mentioned in Subsection 4.2, both the AI agents and the human player can announce a task of the *track and catch tasks* via the CNP interface. When a human player sends an announcement of the *track and catch tasks* to the AI agents, the two AI agents closest to the target will automatically accept that announcement. In other words, a human agent and two AI agents dynamically form a temporary team comprising a three-man cell while sharing the same target. Note that when a human player and AI agents form the same temporary team, the AI agents assume that the human player will take a role as determined by the same flow described above. The human player must perceive her/his own expected role through the task/role symbol of the AI agents in a temporary team in the *catch task*.

6 Experiment

In this section, we discuss what should be taken into account to evaluate the performance of Team A with our method of task decomposition, starting with an overview. We then describe the experimental settings, followed by the results and analysis.

6.1 Overview

In adversarial games such as the Tail Tag game, it is difficult to evaluate the performance of only one of two teams in a competition because most of the metrics such as the win/loss rate depend on the skills of the opponent. To distinguish whether Team A actually performed well as a result of high-level human-AI agent collaboration, it is necessary to ensure that Team B also demonstrated appropriate teamwork. Thus, for Team B, we employed all human players to ensure human-level teamwork and strategy. The win rates and survival rates of each trial enabled us to compare the performance between the human-AI collaboration team (Team A) and the all-human team (Team B). The survival rate is the ratio of the number of agents that kept surviving until the end of the game against the total number of agents on that team.

6.2 Method

To evaluate the performance of our task decomposing approach, we held several matches and compared the win and survival rates. As shown in Table 2, we first held trial #1 as a condition under which the *catch task* was not decomposed, and it included matches of 10 vs. 10 and 20 vs. 20. In this trial, the *catch role* was the only role for the three cooperative agents to catch a target by CNP. Since all AI agents executed the same role, there was no need to declare the intent of the

AI agent to the person, so the task/role symbols and targeting line used by the AI agents to declare intentions were excluded from trial #1. In trial #2, we used the decomposed *catch task*. As described in Section 5, the *catch task* was decomposed into three roles: *catch*, *confine*, and *support*. In addition, the AI agents used the task/role symbols and targeting line to declare their intentions. To evaluate the further scalability of the system in terms of population, we added 30 vs. 30 matches for trial #2.

Table 2: Trial conditions

Trial		**Conditions**		
		Team members	***Decompositions of catch task***	***Task/role symbols and targeting line***
#1	10 vs. 10	Team A: 1 human player and 9 AI agents Team B: 10 human players	Catch Role Only	Not Available
	20 vs. 20	Team A: 1 human player and 19 AI agents Team B: 20 human players		
#2	10 vs. 10	Same as #1, 10 vs. 10	Catch, Confine and Support Roles	Available
	20 vs. 20	Same as #1, 20 vs. 20		
	30 vs. 30	Team A: 1 human player and 29 AI agents Team B: 30 human players		

To avoid dependency on a particular individual, two human players were assigned as a candidate for the Team A human player. In each trial, three matches were held for each one of these two of Team A. As a result, we held a total of six matches at least per trial.

We held three practice games immediately before the experiment in which each team played against a team consisting only of AI agents to ensure that all participants were warmed up and finished familiarizing themselves with controlling their agent. All of the participants were colleagues who worked at the same company as the authors. All of them were adults and experienced engineers but were not involved in this research project. Although the participants of trials #1 and #2 were not all the same, it has been confirmed that there is no correlation between the experience of past participation and performance level in a game when participants have had three practice games in advance [13].

6.3 Results

The win rates for Team A in trial #1 are summarized in Table 3, and the survival rates are shown in Figure 7. The win rate was stable with respect to the population growth. It is also shown that the survival rate of Team A increased and that of Team B decreased as the number of agents increased.

The human players on Team A suggested two improvements to the system in an interview that we held just after the final match of trial #1. The experimental setting of trial #2 was organized in response to these suggested improvements:

- HSI should be improved to make it easier for the person to interact with the AI agents while controlling the player's agent.
- The algorithm of the AI agents' mobility tactics should be improved so that they can share roles such as chasing opponents and move to limit the range of the movable area of the target agent. This should result in better teamwork.

The win rates for Team A in trial #2 are summarized in Table 4, and the survival rates are shown in Figure 8 as well. Compared with trial # 1, the win rate and survival rates for 10 vs. 10 and 20 vs. 20 improved. In comparison, in trial # 2, the win rate and survival rate of Team A got worse as the population grew. This means that the scalability of Team A's performance was inferior to Team B in terms of population growth.

After the very last match of trial # 2, we held an interview the same as trial #1 with the Team A players and got the following comments.

(1) "The task/role symbols and targeting lines were helpful for understanding the intent of the AI agents. This enabled us to decide whether to let the AI agents catch the target or to participate in collaboration."

(2) "In the second half of the game, the AI agents were made to separate and form a locally isolated swarm with a relatively lower population than the agents of Team B around them. As a result, the AI agents were defeated one by one under this locally inferior population situation."

(3) "We tried to gather distributed AI agents by using the retreat task. However, since the retreating AI agent simply tries to move to the target position, they can be captured easily and end up in a vulnerable situation. Therefore, we gave up using the retreat task after some attempts."

Table 3: Results for trial #1

Trial		**Results for Team A**		
		Won	***Lost***	***Win Rate [%]***
#1	10 vs. 10	1	5	17
	20 vs. 20	1	5	17

Table 4: Results for trial #2

Trial		**Results for Team A**		
		Won	***Lost***	***Win Rate [%]***
#2	10 vs. 10	4	2	67
	20 vs. 20	2	4	33
	30 vs. 30	1	5	17

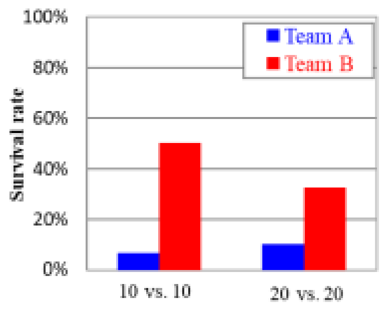


Figure 7: Survival rates for trial #1

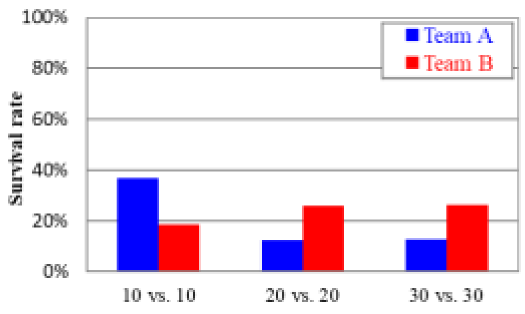


Figure 8: Survival rates for trial #2

6.4 Analysis

We analyze the results and discuss the challenges of our system, new findings, and future improvements to the system.

As we noted above, teamwork is a key to victory. As we noticed in comment (1) of trial #2, the human player could understand the behavior of the AI agents from the task decomposition and declaration of roles. In trial # 2, the human player could understand the role that he/she should take in that situation dynamically. As a result, the win rate and survival rate improved. In particular, in the 10 vs. 10 matches of trial # 2, the win rate was 67%, which means that Team A performed better than Team B.

In comparison, the team B players easily recognized that the Team A agents allocated roles explicitly among them and tried to catch targets in trial # 2. For this reason, as shown in comment (2) of trial # 2, the players on Team B intentionally moved to manipulate the Team-A AI agents to make them separate and kept running away until they had reached an advantageous local situation in terms of number. This observation is supported by Team-B players' interview. One of the reasons that the Team-A agents were made to separate was the defined behavior of the *track task*. As shown in Subsection 5.3, the *track task* is defined as an exception when the target escapes, and it is the simple behavior of going to the target's position. In other words, if the target player keeps running backwards, three AI agents also keep following the target in a row (See Figure 9 as an example). With this behavior, AI agents are easily manipulated, which causes them to be vulnerable.

As a countermeasure against a target that keeps running away, it is effective for trackers to properly surround the target and limit the area where the target can escape. Huang et al. proposed their "*area-minimization strategy*" for the pursuit of a single evader by multiple pursuers [26]. They proved that by moving toward the midpoint of a shared Voronoi boundary between a pursuer and evader, the pursuer is guaranteed to capture the evader in finite time. We are considering applying the area-minimization strategy to the *track task* in future trials.

Finally, we discuss the scalability with respect to population growth. We noticed two things. The first is that the *search task* as an instruction given to the whole of Team A was used at the very beginning of the game only. Second, after the middle of each game, the human players on Team A stuck to catching the closest opponent agent rather than paying attention to the entire team. Since the *catch and track tasks* are performed by three agents, the relative influence of that number of agents assigned to the tasks differs between 10 vs. 10 and 30 vs. 30. In other words, when Team A players were performing a relatively local task such as catching a target, we can say that the global situation got worse as the population grew. As noted in Section 1, we believe that humans are better at making global decisions on the basis of uncertain information than AI. Therefore, it is necessary for a human player to be able to give a global instruction even after the middle of the game. Specifically, as shown in comment (3) for trial # 2, the task of making all AI agents get together while protecting themselves when they were separated should be added. We will verify this gathering task as a *gather task* in future trials.

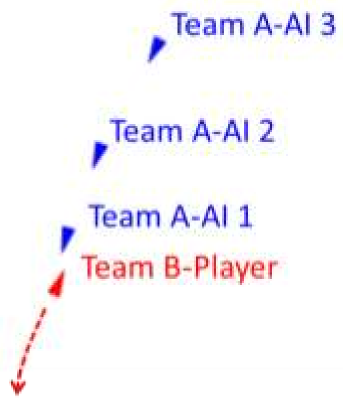


Figure 9: Typical AI agent's behavior in *track task*

7 Conclusion

In this paper, we introduced an architecture and a prototype of a human-multiple agent interface called "human-swarm interaction." The architecture consists of a behavior tree and a contract net protocol as a mechanism for task sharing between humans and AI agents. The use of the behavior tree enables us to know what AI agents are trying to do, while the contract net protocol prioritizes human-AI collaboration without compromising AI agent autonomy. Taking advantage of this architecture, we also presented an approach to decomposing cooperative tasks into appropriate roles. As an example, we decomposed the *catch task* into three roles: *support*, *confine*, and *catch*.

We confirmed that a human player can cooperate with many AI agents at a close degree, similar to a human-only team, through an experiment with a simulated game of Tail Tag. We held matches with up to 29 AI agents and 1 person on one team and 30 people on the other team, both with and without task decomposition to compare the results. The result indicates that decomposing tasks into appropriate roles and sharing the information of the roles that AI agents are currently taking helps humans to understand the AI colleagues' intentions instantly; then, humans can immediately understand the role that they themselves should take. As a result, the win rate and survival rate improved.

In addition, we identified that there is still room for improvement in our system as follows.

- *Area-minimization strategy for track task:*
  AI agents can be easily manipulated and made to separate by human opponents, thus causing the agents to be vulnerable. We can avoid this by using a coordinated pursuit strategy such as moving toward the midpoint of a shared Voronoi boundary between pursuers and an evader.
- *Gather task*:
  We need more scalability with respect to population growth. To achieve this, it is necessary for a human player to be able to give global instructions even after the middle of the game to take advantage of human superiority to AI. Adding the task of making all AI agents get together while protecting themselves when they were separated should be helpful.

We will keep improving not only the above but also the overall system including the AI agent algorithm to achieve more effective teamwork between humans and AI agents on the basis of the results of experiments that we will perform repeatedly.

Acknowledgement

This work was supported by Innovative Science and Technology Initiative for Security Grant Number JPJ004596, ATLA, Japan.

References

[1] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, et al., "Mastering the game of Go without human knowledge," Nature vol. 550, 2017, pp. 354–359.

[2] N. Brown and T. Sandholm, "Superhuman AI for heads-up no-limit poker: Libratus beats top professionals," Science, 2017.

[3] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, et al., "Dota 2 with large scale deep reinforcement learning," arXiv preprint arXiv:1912.06680, 2019.

[4] D. Wang, J. D. Weisz, M. Muller, P. Ram, W. Geyer, C. Dugan, et al., "Human-AI collaboration in data science: Exploring data scientists' perceptions of automated AI," Proceedings of the ACM on Human-Computer Interaction, 3(CSCW):1–24, 2019.

[5] J. Zhu, A. Liapis, S. Risi, R. Bidarra, and G. M. Youngblood, "Explainable AI for designers: A human-centered perspective on mixed-initiative co-creation," in Proc. IEEE Conference on Computational Intelligence and Games, 2018, pp. 458–465.

[6] N. Bard, J. N. Foerster, S. Chandar, N. Burch, M. Lanctot, H. F. Song, et al., "The Hanabi challenge: A new frontier for AI research," arXiv preprint arXiv:1902.00506, 2019.

[7] G. Kasparov, "The chess master and the computer," The New York Review of Books, Vol. 57, No. 2, 2010.

[8] D. C. Dennett, "Cognitive Wheels: The Frame Problem of AI," Language and Thought, vol. 3, 2005.

[9] D. Gunning, "Explainable artificial intelligence (XAI)," Defense Advanced Re-search Projects Agency (DARPA), 2017.

[10] A. Stentz, C. Dima, C. Wellington, H, Herman, and D. Stager, "A system for semi-autonomous tractor operations in Autonomous robots," Autonomous Robots, Vol. 13, 2002, pp. 87-104.

[11] S. Balakirsky, S. Carpin, A. Kleiner, M. Lewis, A. Visser, J. Wang, and V. A. Ziparo, "Towards heterogeneous robot teams for disaster mitigation: Results and performance metrics from robocup rescue," Journal of Field Robotics, 24, 2007, pp. 943-967.

[12] J. Li, and H. Liu, "Design Optimization of Amazon Robotics. Automation," Control and Intelligent Systems, Vol. 4, No. 2, 2016, pp. 48-52.

[13] A. Kolling, P. Walker, N. Chakraborty, K. Sycara, and M. Lewis, "Human interaction with robot swarms: A survey," IEEE Transactions on Human-Machine Systems, Vol. 46, No. 1, 2016, pp. 9-26.

[14] A, B and C (anonymity for a blind review), “Architecture and interface for collaborating with a group of agents in an adversarial game,” Proceeding of a conference, 2020. (published)

[15] M. Colledanchise, and P. Ogren, “Use of BTs in robotics and AI, in Behavior Trees in robotics and AI: An introduction,”, 1st Ed., CRC Press, 2018.

[16] R.G. Smith, “The Contract Net Protocol: High-level communication and control in a distributed problem solver,” In IEEE Transactions on Computers, Vol. C-29, No. 12, December 1980, pp. 1104-1113.

[17] M. Tambe, “Towards flexible teamwork,” Journal of Artificial Intelligence Research, Vol 7, 1997, pp. 83-124.

[18] R. Nair, M. Tambe, and S. Marsella, “Role allocation and reallocation in multiagent teams: towards a practical analysis,” Proceedings of the second international joint conference on Autonomous agents and multiagent systems, 2003, pp. 552-559.

[19] P. Stone and M. Veloso, “Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork,” Artificial Intelligence, Vol. 110, Issue 2, 1999, pp. 241-273.

[20] M. B. Dias, R. Zlot, N. Kalra and A. Stentz, “Market-based multirobot coordination: A survey and analysis,” Proceedings of the IEEE, Vol. 94, No. 7, 2006, pp. 1257-1270.

[21] D.V. Pynadath and M Tambe, “Multiagent teamwork: analyzing the optimality and complexity of key theories and models,” Proceedings of the second international joint conference on Autonomous agents and multiagent systems, 2002.

[22] C. J. Cai, J. Jongejan, and J. Holbrook, “The effects of example-based explanations in a machine learning interface,” Proceedings of the 24th International Conference on Intelligent User Interfaces, 2019, pp. 258–262.

[23] P. Stone, R. S. Sutton, and G. Kuhlmann, “Reinforcement learning for RoboCup-soccer keepaway,” Adaptive Behavior, Vol. 13, 2005, pp.165-188.

[24] S. Ontañon, G. Synnaeve, A. Uriarte, F. Richoux, D. Churchill, and M. Preuss, “A Survey of Real-Time Strategy Game AI Research and Competition in StarCraft,” IEEE Transactions on Computational Intelligence and AI in games, IEEE Computational Intelligence Society, 2013, pp. 1-19.

[25] M. Jaderberg, W. M. Czarnecki, I. Dunning, L. Marris, G. Lever, A. G. Castañeda, et al., “Human-level performance in first-person multiplayer games with population-based deep reinforcement learning,” Science 364, 2019, pp. 859-865.

[26] H. Huang, W. Zhang, J. Ding, D. Stipanovic, and C. Tomlin, “Guaranteed decentralized pursuit-evasion in the plane with multiple pursuers,” Proceedings of IEEE Conference on Decision and Control, 2011, pp. 4835–4840.