Int. J. Appl. Math. Comput. Sci., 2023, Vol. 33, No. 4, 553–568
DOI: 10.34768/amcs-2023-0040

ASTS: AUTONOMOUS SWITCHING OF TASK–LEVEL STRATEGIES

XIANCHANG WANG[a,b,c], BINGYU LV[a,b], KAIYU WANG[a,b], RUI ZHANG[a,b,*]

[a] Institute of Computer Science and Technology
Jilin University
No. 2699 Qianjin Street, 130021 Changchun, Jilin, China
e-mail: rui@jlu.edu.cn

[b] Key Laboratory of Symbolic Computation and Knowledge Engineering of the Ministry of Education
Jilin University
No. 2699 Qianjin Street, 130021 Changchun, Jilin, China

[c] Research and Development Department
Chengdu Kestrel Artificial Intelligence Institute
No. 17 Dayu East Road, Deyuan Jingrong Town, Pidu District, 611136, Chengdu, Sichuan, China

Autonomous coordination of multi-agent systems can improve the reaction and dispatching ability of multiple agents to emergency events. The existing research has mainly focused on the reactions or dispatching in specific scenarios. However, task-level coordination has not received significant attention. This study proposes a framework for autonomous switching of task-level strategies (ASTS), which can automatically switch strategies according to different scenarios in the task execution process. The framework is based on the blackboard system, which takes the form of an instance as an agent and the form of norm(s) as a strategy; it uses events to drive autonomous cooperation among multiple agents. A norm may be triggered when an event occurs. After the triggered norm is executed, it can change the data, state, and event in ASTS. To demonstrate the autonomy and switchability of the proposed framework, we develop a fire emergency reaction dispatch system. This system is applied to emergency scenarios involving fires. Five types of strategies and two control modes are designed for this system. Experiments show that this system can autonomously switch between different strategies and control modes in different scenarios with promising results. Our framework improves the adaptability and flexibility of multiple agents in an open environment and represents a solid step toward switching strategies at the task level.

Keywords: task-level, autonomous switching strategies, blackboard system, norm.

1. Introduction

In recent years, fires have become a major threat to humans and property worldwide. It has the potential to cause billions of dollars' worth of economic damage and kill thousands of people every year, resulting in catastrophic social, economic, and environmental impacts. Considering the massive devastation caused by fires, the use of unmanned devices to respond to emergencies has become extremely important. Collaborative or dispatching methods for the existing multi-agent systems (MASs) have demonstrated their performance (Li *et al.*, 2022). In an MAS, multiple agents are primarily controlled through supervised learning (Landgren *et al.*, 2021; Tan *et al.*, 2022). Multiple agents primarily learn from specific scenarios that may be encountered. These methods are based on the collected data set for learning specific behaviours to determine the optimal reaction. Fires occur in a variety of environments, such as forests, grasslands, mountains, roads, etc. When multiple agents encounter a certain scenario and are unable to make a decision, a large amount of data is required to support learning. However, poor data quality, due to incompleteness, inaccuracy, and bias, leads to poor judgments and decision making. Therefore, new solutions are required to maximize the overall flexibility of the MAS to react in different scenarios.

In the MAS, common accident scenarios are predicted in advance by experts. The existing research focuses mainly on reactions or dispatching in specific scenarios (Wu *et al.*, 2018; Wang *et al.*, 2020; Bai

*Corresponding author

et al., 2021; Sengupta and Yasser Mohammad, 2021; Roy *et al*., 2018; Liu *et al*., 2022; Sun and Liu, 2021), whereas little attention has been paid to task-level coordination (Kanazawa *et al*., 2021). Wu *et al*. (2018) studied the dual-objective rescue vehicle dispatching problem of a multi-point forest fire to minimize the total fire extinguishing time and the amount of fire equipment required. However, this method lacks foresight and feedback regarding long-term plans, which may lead to short-sighted decision making. In addition, in some special cases, heuristic methods may produce poor efficiency and answers,which will affect the dispatching decision. Therefore, Wang *et al.* (2020) proposed a switching strategy based on real-time wind process prediction, which mainly dynamically adjusts strategies according to the wind type. Such a method-switching strategy must predict the future and conduct cluster analysis. Bai *et al*. (2021) studied the periodic switching strategy method, which is different from the conventional methods. They also proposed an operation optimization model to obtain the best switching times and corresponding time points. When an exception occurs in the MAS, the periodic switching strategy may not react in an unexpected situation. Sengupta and Yasser Mohammad (2021) explored a framework for selecting, switching, or combining strategies. Roy *et al*. (2018) proposed a switching strategy framework of region-based shape controllers for swarm robots to overcome the traditional obstacle avoidance problem. Kanazawa *et al*. (2021) investigated a new method to adaptively select two objective functions according to the current operating area of the robot, which can improve the safety of workers. However, this approach does not allow the system to evolve under unexpected circumstances.

The norms for controlling multiple agents within an MAS depend on the expertise of the experts. However, the experts cannot predict all unexpected scenarios. The designed norms are not adequate for dealing with all scenarios and must evolve to adapt to changing scenarios (Zhao *et al*., 2017; Wang *et al*., 2020c). Sengupta and Yasser Mohammad (2021) studied a component that periodically replaces old strategies with new or better ones for self-enhancement. Zhao *et al*. (2017) presented a framework of software adaptation, norm generation, and norm evolution based on reinforcement learning. This framework automatically learns during the offline stage and automatically evolves during the online stage. According to Wang *et al.* (2020c), mutation is the primary strategy for norm evolution. Their method used the operator ‘crossover’ to evolve norms. The effectiveness of the evolutionary approach was verified using a sweeping robot that performed the sweeping task. However, this method does not investigate the phenomenon wherein evolved norms are not easily triggered when exceptions occur for many norms.

For multiple agents to autonomously switch between different strategies in different scenarios, this study proposes a framework for autonomous switching of task-level strategies (ASTS). In ASTS, multiple agents cooperate to complete the overall goal. In the ASTS framework, multiple agents perform a collection of actions in one state and then change to a new one. Subsequently, to prove the effectiveness of ASTS, a fire emergency reaction dispatching (FERD) system based on the ASTS framework is established. This dispatch system is built for fire events and includes five types of strategies and two control modes. There are always situations in the operation of this system that are not considered by the norms. Under unknown conditions, the system can implement basic motion safety guidelines. They perform their activities using these basic safety guidelines in the case of multiple agents having accidents or not satisfying the preconditions of the norms.

With the system operation, various sudden and unexpected situations can be considered in the future. Simultaneously, various emergency measures are planned and implemented to avoid a short-term extension of the accident and delayed firefighting. This study validates the FERD system by applying some experiments to the domain of extinguishing fireworks. Finally, the effectiveness of this framework is proven, and the experimental results are analysed. The main contributions of this study are summarized as follows:

- This paper presents a framework for autonomous switching strategies to achieve the overall goal, which is at the task level. The strategies are used to drive the autonomous collaboration of multiple fire equipment according to different scenarios. The switching between different strategies proves that the framework is agile and extensible, benefiting from the framework.

- The switch and strategy modules are designed for the FERD system, which includes two switch modes and five types of strategies. They reflect the (non) autonomy, collaboration, and exception-handling capabilities in different scenarios. These strategies are obtained based on the environmental model, task requirements, task execution, and dynamics of the agent, as well as the processing of the collected environmental information. They can be divided into two levels: norm-level and agent-level strategies.

- This study provides an in-depth analysis of strategies, and studies the relationship between components of norms and the properties of knowledge source (KS) in generic blackboard open source (GBBopen). The FERD system is implemented based on GBBopen, indicating its expandability. This paper also improves the control shell of GBBopen with a few improvements, primarily improving the properties of knowledge source (KS) and adds a slot to KS, *postcondition-function*. By taking the Chengdu Kestrel Artificial Intelligence Institute's IDL-Mapping tool (KIS-CORBA) as the cross-language data communication protocol, obtaining real-time information with multiple sensors, and publishing or subscribing to information through the serial port of the robot operating system (ROS), which shows the openness of the FERD system.

The remainder of this paper is organized as follows. Section 2 discusses the related work. The framework and terms of ASTS are given and the FERD system proposed in Section 3. In Section 4, the experiments to verify the effectiveness of the FERD system are presented; subsequently, the experimental results are presented and analyzed. Finally, Section 5 summarizes the significance of the proposed framework and system, and discusses future work.

2. Related work

Related studies have investigated MAS intelligence in terms of collaboration, decision making, communication, and learning. Among these, decision making is the core research problem of MAS autonomous collaboration. Autonomous collaboration is a popular topic in MAS research.

2.1. MAS decision making.

Tan *et al.* (2022) designed an interactive framework to motivate each agent to align personal goals with system goals. They also proposed a novel online learning algorithm for MASs in dynamic environments. The algorithm uses partial, delayed, and noisy state information as well as reward signals for learning. Lin *et al.* (2018) proposed a novel Q-learning-based adaptive modulation-switching strategy to select the appropriate strategy. Landgren *et al.* (2021) designed a dynamic, consensus-based, distributed estimation algorithm. The algorithm separately considers unconstrained and constrained reward models and estimates the average reward for each arm for selective learning. The methods proposed by Tan *et al.* (2022), Lin *et al.* (2018) or Landgren *et al.* (2021) make decisions regarding MASs via reinforcement learning. The agents of Tan *et al.* (2022), Lin *et al.* (2018) or Landgren *et al.* (2021) require an exploration of the action space, respectively. They can quickly adapt to new environments to determine optimal strategies and actions. However, these methods require considerable amounts of time and computational resources. There are also situations in which some strategies are not selectable, to the detriment of discovering more valuable ones. The difference between this study and their work is that the agents in the FERD system do not need to explore the action space. As multiple agents are tasked within a fixed environment, the strategy or action has been predefined. A norm is successfully triggered when the subscribed events are published and the preconditions of the norm are satisfied. Multiple agents are executed according to the predefined actions.

The existing research is based on a collected data set to learn specific behaviours to determine an optimal action. Hook *et al.* (2021) focused on learning actions in an MAS. These agents learn primarily by simulating scenarios that may be encountered during autonomous navigation. Ma *et al.* (2020) investigated a nonparametric closed-loop behaviour learning method for MAS motion planning. Data-driven learning offers objectivity, validity, iterability, visualisation, and comprehensiveness; however, it requires a large amount of data to support learning or prediction. Data incompleteness, inaccuracy, and bias lead to poor judgments and decisions. This study differs from their work in that it is event-driven to control multiple agents. Responding promptly when an event occurs prevents polling; thus, the reaction speed and efficiency of the FERD system can be improved and the ability of real-time processing can be realized. The FERD system can also add, remove, and modify events, strategies, or control modes dynamically as needed, thereby increasing its flexibility.

Ming *et al.* (2022) proposed a strategy selection algorithm, which is modelled as a game. Zhou (2021) proposed a novel optimal control framework for MASs which introduces the mean-field game theory. The above approaches require modelling the entire game process. However, this information may be incomplete or inaccurate, which may lead to instability and performance degradation of the algorithm. In contrast, the approach in this study relies on the current agent state and blackboard content. Changes in the blackboard content must be monitored in real time, which mainly includes data on the blackboard, task status, and execution results. By monitoring the contents of the blackboard, their failures and abnormalities are detected in time to provide a basis for subsequent strategy selection.

2.2. MAS decision making in different scenarios.

Some existing studies are similar to ours, such as combining different scenarios with MAS decisions to enable the MAS to switch decisions between multiple scenarios (Sengupta and Yasser Mohammad, 2021; Roy *et al.*, 2018; Wang *et al.*, 2020b; Nimmolrat *et al.*, 2021). Roy *et al.* (2018) proposed several obstacle avoidance strategies, which are designed to overcome the traditional obstacle avoidance problem. The robots form a group and move within the circular area of the initial virtual structure. Simultaneously, they move as various specific structures, such as a triangular structure. The virtual circle can be allowed to shrink to a certain limit to avoid static or dynamic obstacles. Wang *et al.* (2020b) studied the configuration of rescue vehicles and dispatch strategies in highway emergencies. First, traffic accidents were classified into categories. Subsequently, based on the research and statistics, the type and number of rescue vehicles required were determined. Finally, three dispatching methods are proposed and validated to enable rescue vehicles to reach traffic accident points as soon as possible. Nimmolrat *et al.* (2021) proposed linking the dispatch centre to the rescue team to assist with the mission. The first-aid centre receives information from the geolocation system to determine the location of the patient. The efficiency of the prehospital process is improved, which also helps paramedics and rescue teams allocate medical resources and deal with emergencies in a timely manner.

Inspired by Wang *et al.* (2020c), we extend this study to a task-level setting and use an evolutionary algorithm for norm evolution. This minimizes the impact of anomalous results on the FERD system and improves its robustness. In addition, the proposed autonomous switching strategy framework can be developed for different contexts and states, thereby improving the system availability; the number of strategies can be dynamically increased or decreased based on the actual demand, flexibly responding to different needs and demonstrate the versatility of the proposed approach.

3. Task-level autonomous switching strategy

This section introduces the basic knowledge of this study and proposes an ASTS framework. The system for applying this framework to extinguishing fire scenarios is presented in the remainder of this section.

3.1. Preliminaries.

Blackboard system. The blackboard system (BBS) is a model for solving unstructured problems such as intelligent decision-making and intelligent control, which is driven by events. It comprises a KS, blackboard (BB), and control shell (CS). The KS is similar to a field specialist. The BB displays the data in real time. The CS selects the appropriate KS according to the event. Many articles on BBS are quite detailed (e.g., Shin *et al.*, 2018).

Generic blackboard open source. Based on the different characteristics of the application problems, BBS has many different forms. The software GBBopen is developed and implemented in the common list processing (Common Lisp) language environment. GBBopen is open-source software, and users can improve it by modifying the source code. An expert system based on the norm-base developed by Clips[1] is a data-driven program. Unlike Clips, GBBopen does not require a data abstraction design; therefore, the implementation of a function may be completed quickly during development.

Topic of the robot operating system. The topic of communication frameworks in ROSs consists of three parts: the speaker, the listener, and the master. The data are then transferred from the talker to the listener (Oguz-Ekim *et al.*, 2020).

3.2. Overall framework.

In this subsection, a framework for ASTS is proposed for multiple agents to switch strategies autonomously according to different scenarios. The overall framework, terms, algorithms, and implementation of the framework were introduced.

3.2.1. Terms for ASTS.

An ASTS is described in the form of the triple consisting of a set of norms N, state S, and event E,

$$\text{ASTS} = \langle N, S, E \rangle. \tag{1}$$

Norm N. $N = \{r_1, r_2, \dots\}$ represents a set of norms, which can be used to guide and constrain the behaviour of the agent, ensuring the normal operation of the ASTS. Norm r_i is expressed as $r_i = \{Tri_i, Act_i, Exp_i\}$.

- Tri_i is the trigger condition and is used to determine if the associated actions can be performed. Tri_i comprises a series of logical judgment expressions denoted as $Tri_i = \{tri_{i_1}, tri_{i_2}, \dots\}$; tri_{i_j} is a logical judgment expression.
- Act_i is the execution action. Act_i consists of a series of methods or actions, which can be expressed as $Act_i = \{act_{i_1}, act_{i_2}, \dots\}$; act_{i_j} is a method or an action.
- Exp_i is the expected result to verify the result of Act_i. Exp_i consists of a series of expected conditions, which can be expressed as $Exp_i = \{exp_{i_1}, exp_{i_2}, \dots\}$; exp_{i_j} is also a logical judgment expression.

State S. The data are read by state S. State S represents a set of instantaneous states denoted by $S = \{s_1, s_2, \dots\}$. The elements in state S is arranged in chronological order. The instantaneous state s_i can have multiple expressions, which are represented as $s_i = \{s_{i_1}, s_{i_2}, \dots\}$.

Event E. $E = \{e_1, e_2, \dots, e_i\}$ represents a set of instantaneous events sorted in chronological order. e_i denotes the i-th instantaneous event. Here $e_i = \{e_{i_1}, e_{i_2}, \dots\}$ indicates that e_i is composed of multiple sub-events. When an event e_i occurs, the instantaneous state s_j is read to determine the Tri_k of r_k.

[1] https://www.clipsrules.net.

Definition 1. (*Norm, event, and state*) Equation (1) describes the components of the ASTS. Examples of norms, events, and states are presented in Example 1.

Example 1. (*Norm, event, and state*) A robot named Anna starts working at 8:00 every day. Anna maintains a safe distance from the object in front of it and must obey norms r_a and r_b listed in Table 1. The distance between Anna and the object ahead is expressed as D. The safe distance is set to 1 m. If D is greater than 1 m and the power is greater than 20% when e_1 occurs, then Anna moves forward. To verify the result Act_a of the norm r_a, we expect D to be equal to 1 m. If D is less than 1 m and the power is less than 20% when e_1 occurs, Anna stops. The instantaneous events and states are listed in Table 2. ♦

Definition 2. (*Triggered norm*) When e_i occurs, the instantaneous state s_i is read. If all logical judgment expressions in Tri_x are satisfied, then norm r_x is referenced as the 'triggered norm' at instantaneous state s_i. Examples of triggered norms are provided in Example 2.

Example 2. (*Triggered norm*) At 8:00, an instantaneous event e_1 occurs. Tri_a is satisfied at the instantaneous state s_1; then Anna starts moving forward. Norm r_a is denoted as the 'triggered norm'. ♦

Definition 3. (*Obeyed norm and mutated norm*) The transient state of s_i reaches s_j after the execution of Act_x. If all logical judgment expressions in Exp_x are satisfied at s_j, then norm r_x is denoted as an 'obeyed norm.' Otherwise, norm r_x is denoted as a 'mutated norm.' Examples of the obeyed norm and the mutated norm are shown in Example 3.

Example 3. (*Obeyed norm and mutated norm*) Anna starts executing Act_a in s_1 and the instantaneous state from s_1 reaches s_2. Exp_a is satisfied at instantaneous state s_2. Therefore, norm r_a is denoted as an 'obeyed norm.' At 8:02, the norm r_a is triggered. Anna starts executing Act_a in s_2 again and the instantaneous state from s_2 reaches s_3. Exp_a is not satisfied at instantaneous state s_3. Therefore, norm r_a is denoted as a 'mutated norm.' ♦

Definition 4. (*Execution path of norms*) This path is formed by recording triggered norms, instantaneous events, instantaneous states, results of triggered norms, and other data. In this path, the norm is represented as a circle, whereas an instantaneous state is presented as a block. The result of a triggered norm is expressed as a directed line segment. The 'obeyed norm' is marked by a solid line and the 'mutated norm' is marked by a dotted line. Examples of the execution paths of the norms are presented in Example 4.

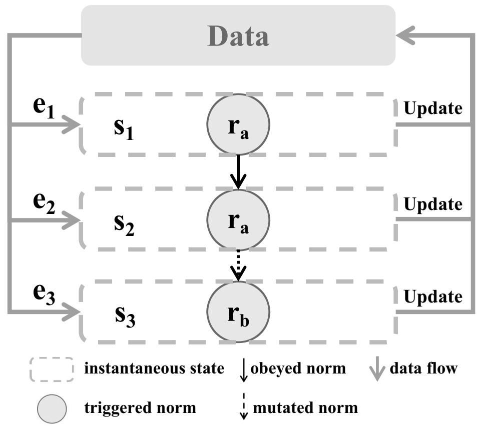


Fig. 1. Execution path of norms.

Example 4. (*Execution path of norm*) From Fig. 1, norm r_a is triggered at s_1, then Act_a is executed. The result of Exp_a indicates that the norm r_a is the 'obeyed norm.' Norm r_a is triggered at s_2 again. The result of Exp_a indicates that the norm r_a is a 'mutated norm.' Norm r_b is triggered at s_3. ♦

Definition 5. (*Transition state*) When e_i occurs, norm r_x is triggered at s_i, and Act_x is executed. This process causes the instantaneous state to change from s_i to s_j, which can be expressed as

$$s_i \xrightarrow{r_x} s_j, \tag{2}$$

where '→' denotes an instantaneous state changed from s_i to s_j as a result of triggering r_x. Examples of the transition states are shown in Example 5.

Example 5. (*Transition state*) From Fig. 1, the instantaneous state changed from s_1 to s_2 as a result of triggering r_a. The instantaneous state changes from s_2 to s_3 as a result of triggering r_a. These processes are expressed as

$$s_1 \xrightarrow{r_a} s_2 \xrightarrow{r_a} s_3. \tag{3}$$

♦

Definition 6. (*Strategy*) Norms describe the actions that an agent should take in a given situation. Norms are the basis of strategies and strategies are action plans based on norms. Norms and strategies are often both interrelated and interdependent. Examples of strategies are given in Example 6.

Example 6. (*Strategy*) In Table 1, norms r_a and r_b form a goal-following strategy. Norm r_c forms an obstacle-avoidance strategy. ♦

Definition 7. (*Switching norms or strategies*) When e_i occurs, norm r_x is triggered at s_i. Then e_j occurs and r_y is triggered at s_j. The norm switches from r_x to r_y.

Table 1. Example of a norm.

Norm	Trigger	Action	Expectation
r_a	$D > 1$ m, $power > 20\%$	Anna forward	D = 1 m
r_b	$D < 1$ m, $power > 20\%$	Anna stop	D = 1 m
r_c	$D < 0.5$ m, $power > 20\%$	Anna backward	$D > 0.5$ m

Table 2. Recording of instantaneous events and states.

Time	Instantaneous event	Instantaneous state
8:00	Robot open Subscribe to Anna's position	D = 2.1 m, $power$ = 28%
8:02	Subscribe to Anna's position	D = 1.5 m, $power$ = 24%
8:05	Subscribe to Anna's position	D = 0.9 m, $power$ = 21%
8:10	Subscribe to Anna's position	D = 0.8 m, $power$ = 19%

Multiple switching norms accumulate when one norm is switched to another. Examples of the switching norms and strategies are presented in Example 7.

Example 7. (*Switching norms or strategies*) From Fig. 1, when e_1 occurs, norm r_a is triggered at s_1. Then, e_2 occurs, and the norm r_a is triggered at s_2. Finally, e_3 occurs, and the norm r_b is triggered at s_3. The norm switches from r_a to r_a and eventually to r_b. ♦

Based on the above description, this study summarizes the algorithm and pseudocode (Algorithm 1) of the ASTS as follows:

i. Input N, S, E, T;

ii. Initialisation: $Path = \emptyset, Res_{Act} = \emptyset, Res_{Exp} = \emptyset, i = 0, j = 0, k = 0$.

iii. If an instantaneous event $e_i \in E$ occurs, the ASTS data are read by the transient state $s_j \in S$.

iv. Check Tri_k of norm N.

v. If $Tri_k = True$, norm r_k is triggered; perform Act_k and check Exp_k.

vi. The results for Act_k and Exp_k are restored as Res_{Act} and Res_{Exp}, respectively.

vii. $s_i, e_j, r_k, Res_{Act}, Res_{Exp}$ denote the restored paths.

viii. $j++, i++$.

ix. Until $i > T$, exit ASTS.

3.3. Implementation for ASTS.

This subsection proposes a FERD system to respond to fire extinguishing tasks based on the ASTS framework. The FERD system aims to perform extinguishing fire tasks using multiple heterogeneous types of agents. Subsequently, the details of the FERD system are introduced.

3.3.1. Implementation system.

The FERD system consists of a set of tasks M, agent A, and norm N, based on the ASTS.

- **Task M.** The FERD system is applied to handle fire extinguishing tasks, task $M = \{m_1, m_2, \dots\}$. Each task m_i comprises a series of attributes denoted as $m_i = \{m_{i_1}, m_{i_2}, \dots\}$.

- **Agent A.** A certain number of agents is needed to achieve the ultimate goal. Each agent has a unique identification number, such as a_1, a_2, so A = $\{a_1, a_2, \dots\}$.

- **Agent norm N'.** The norm set for the agent is N' = $\{r'_1, r'_2, \dots\}$, where norm r'_k is expressed as r'_k = $\{Tri'_k, Act'_k, Exp'_k\}$. The internal elements of Tri'_k, Act'_k, Exp'_k and the internal elements of Tri_k, Act_k, Exp_k are consistent.

3.3.2. Constituent.

The FERD system operates based on the GBBopen platform. In the FERD system, a set of multiple KSs is expressed using the strategy of agents. Each agent exists in the form of an instance. The CS selects the appropriate KS for the multiple agents to switch. After the strategy is executed, the data on the BB are changed until the fire extinguishing tasks are completed. Certain properties of the agent are obtained from external sensors. The components of the FERD system are introduced in detail.

1. *Blackboard.* The agent in the FERD system is defined by a unit class. The unit class is the base class for all classes defined by GBBopen. The task is defined as a class. Each agent type is defined as a class. Each class has several properties. The attribute value of each agent is either defined directly (e.g., name) or obtained through the external interface of the sensor (e.g., position coordinates obtained by the

Algorithm 1. Autonomous switching of task-level strategies (ASTS).

Require:
N, a set of norms
S, a set of instantaneous states
E, a set of instantaneous events
T, a time threshold
Res_{Act}, a result of Act_i
Res_{Exp}, a result of Exp_i
Ensure:
$Path$ is a path of triggered norms

while $i \leq T$ **do**
  s_i←data;
  if $e_j \neq \emptyset$ **then**
    for each $r_k \in N$ **do**
      Assess Tri_k;
      if Tri_k = True **then**
        Res_{Act}←do Act_k;
        Assess Exp_k;
        if Exp_k = True **then**
          Res_{Exp}←obeyed norm;
        else
          Res_{Exp}←mutated norm;
        end if
        $Path$←add($s_i, e_j, r_k, Res_{Act}, Res_{Exp}$);
      end if
      $k++$;
    end for
    $j++$;
  end if
  $i++$;
end while
return $Path$;

positioning sensor). Each instance of an agent and its attribute values are displayed on theBB.

2. *Knowledge source.* The KS is a core component of the FERD system. Each knowledge source activation (KSA) is an instance of a KS. In GBBopen, the attributes of the KS are in one-to-one correspondence with the agent norm N'. In this study, the properties of the KS are extended to 'trigger-events', 'precondition-function', 'execution-function', and 'postcondition-function.' The relationship between N' and the KS can be formally described as follows: ks_i = {event, pre, exe, rating, post}; ksa_i={rating, exe, post}; $Tri'_i = ks_i$.event and ks_k.pre, $Act'_i = ks_i$.exe, $Exp'_i = ks_i$.post.

KSs are divided into two submodules: the *switch module* and the *strategy module*. In the strategy module, the strategies are obtained based on the dynamics of the robot and the processing of the collected environmental information. In the switch module, the agent switches strategies in the strategy module according to the different control modes. Figures 2 and 4 show the classifications in the strategy module and the switch module, respectively.

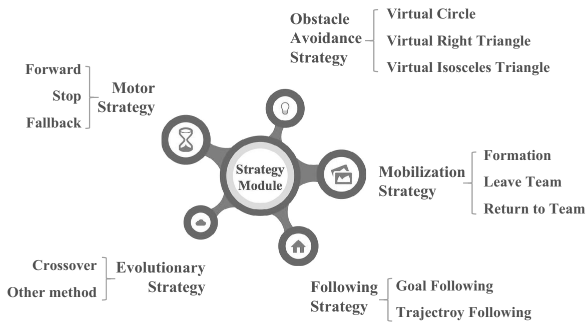


Fig. 2. Classification in the strategy module.

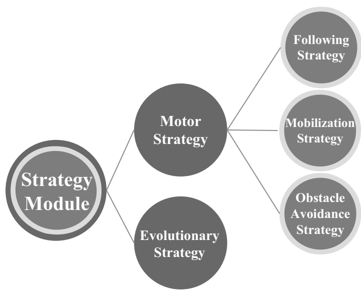


Fig. 3. Relationship of strategies in the strategy module.

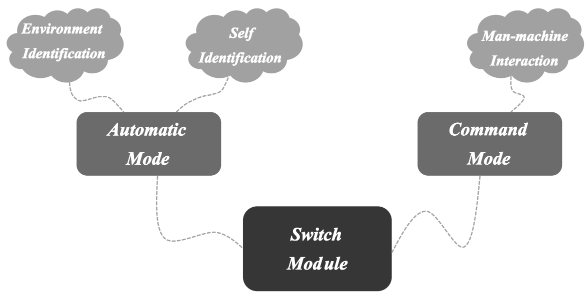


Fig. 4. Classification in the switch module.

1. *Classification of the switch module.* Multiple agents can switch strategies in the strategy module depending on the different control modes. The control modes are divided into an *automatic mode* and a *command mode*.

   (a) *Automatic mode.* In different scenarios, multiple agents can autonomously switch strategies based on their own identification or the environmental identification. This control mode avoids the problem of an optimal strategy not being selected due to insufficient manual experience. For instance, Anna autonomously switches the norms in Table 1 according to the comparison results of D and the safety distance in the automatic mode.

(b) *Command mode.* In different scenarios, multiple agents in the command mode can switch strategies based on the user's needs or external session guidance. For example, Anna switches the norms in Table 1 according to the user commands in the command mode.

2. *Classification of the strategy module.* These strategies are obtained based on the task requirements, fire extinguishing task execution, dynamics of the agent, and processing of the collected environmental information by experts from different fields. These mainly include the *following strategy*, *motor strategy*, *evolution strategy*, *obstacle avoidance strategy*, and *mobilization strategy*. Figure 3 shows the relationship of strategies in the strategy module. The relationship between these strategies can be divided into two levels: norm-level and agent-level. The motor strategy is the basis of the agent movement. The following strategy, and obstacle avoidance strategies are implemented using various norms in the motor strategy. They belong to the agent-level. The evolution strategy is used for norm evolution, which belongs to the norm-level.

   (a) *Movement strategy.* The movement strategies are divided into *forward*, *stop*, and *fallback*. 'Forward' allows multiple agents to move forward rapidly. This strategy is suitable for specific scenarios such as patrol, transportation, etc. 'Stop' can make multiple agents stop precisely at a specified position, thereby effectively avoiding collisions and other unexpected situations. 'Fallback' can cause multiple agents to move backward according to the specified path. Appropriate motion commands can be selected as required for different task scenarios.

   (b) *Following strategy.* In this study, we developed two strategies for agents to arrive at fire scenarios rapidly. These are *trajectory following* and *goal following*. 'Goal following' can adaptively adjust the direction and velocity of agents according to changes in the position and motion state of the target or fire. 'Trajectory following' can precisely repeat the fire extinguishing task according to a pre-set trajectory.

   (c) *Mobilization strategy.* In different task scenarios, multiple agents can choose the appropriate control mode as required. The FERD system provides three strategies: *formation*, *leave team*, and *return to team*. 'Formation' enables multiple agents to form an orderly queue of movements to collaborate on tasks. This strategy can effectively avoid collisions and mutual interference, thereby improving security. This can also improve the efficiency and speed of task execution. 'Leave team' allows multiple agents to move and explore freely without constraints, explore new environments and goals, and discover new information and resources. 'Return to team' can allow the agent to return to the formation position precisely according to the specified path and method, effectively avoiding collisions and other unexpected situations for this agent.

   (d) *Obstacle avoidance strategy.* The FERD system provides three obstacle avoidance strategies: *virtual circle*, *virtual right triangle*, and *virtual isosceles triangle*. Multiple agents can adjust their path and direction of motion based on changes in the obstacle location. 'Virtual right triangle' and 'virtual isosceles triangle' strategies usually allow for movement directly along a line path without requiring extra time to move around obstacles. 'Virtual circle' typically allows for movement around obstacles while maintaining an overall streamlined shape.

   (e) *Evolutionary strategy.* When the FERD system is operational, the results of the same execution norm may differ for an agent. However, experts cannot predict all states in the system. Therefore, the norms of the system must be evolved. The FERD system provides two evolutionary strategies for autonomously evolving norms: '*crossover*' and '*other method*'. 'Other method' strategy is not described in this paper.

In 'crossover', the expectation of one mutated norm enrich the trigger of another mutated norm. The crossover algorithm is as follows:

i. The path is entered, which is obtained from Algorithm 1.

ii. Initialisation: the mutated norm (MN) $= \emptyset$.

iii. For each $path_i \in path$, check the mutated norm r_j. If a mutated norm r_j occurs, it is assigned to MN.

iv. If the length of MN is greater than 2, for each $mn_k \in MN$ ($k \neq 0$), the expected condition of the exception mn_k takes the antonym and is added to the trigger condition of mn_0.

v. Output mn_0.

In particular, we assume that the norm execution sequence obtained by the path is r_a, r_b and r_c. First, all mutated norms are detected from this path, which are the norms r_a and r_c. The exception of r_c is detected to be caused by the exception of r_a. Second, we extract the mutated expectations of norm r_c, namely, exp_{c_3}, exp_{c_5}, $exp_{c_6} \in Exp_c$. They are then considered as antonyms, namely, not exp_{c_3}, not exp_{c_5}, and not exp_{c_6}, which is the new knowledge of this system. Next, new knowledge is added to the trigger of another mutated norm through the crossover operator. Finally, an improved trigger is obtained: $Tri_a = \{tri_{a_1}, tri_{a_2}, \ldots,$ not exp_{c_3}, not exp_{c_5}, not $exp_{c_6}\}$.

3 *Control shell.* The CS can manage the triggering of various KSs and is responsible for dispatching KSs in the FERD system. The running process of the CS is as follows: Initialization: Various KSs, agents, T and M exist within the FERD system. When an instantaneous event e_x occurs in the system, the CS determines whether it is consistent with the $ks_i.event$ of $ks_i \in KS$. If they are consistent, the instantaneous state s_y reads the data of agents from the BB and the function $ks_i.pre$ is executed. The variable value of $ks_i.pre$ is provided by s_y. If the return value of the function $ks_i.pre$ is true, then ks_i is triggered and generates an instance ksa_i, which stores a list of 'pending-ksas.' Then, $ksa_k.exe$ in 'pending-ksas' with the highest level is executed. The data of the agents on BB are changed and $ksa_k.post$ is executed. The CS triggers various norms based on event changes to achieve fast switching strategies. When $M = \emptyset$ or $t > T$, the FERD system ends its operation. The output path is the execution sequence for KSs.

4. Experimental results and analysis

In this section, to verify the effectiveness and switchability of the FERD system, experiments were designed for multiple agents in response to emergency events. Experiments were performed on a series of fire extinguishing tasks as a specific application scenario for the proposed FERD system. Multiple agents can automatically or passively switch between different strategies during their fire extinguishing tasks.

First, the experimental design is introduced, including the experimental scenario, settings, platform, equipment, contents, and evaluation. The steps of the experiment are then described. Finally, the experimental results from performing the tasks are presented and analyzed.

4.1. Experiment design.

4.1.1. Experimental scenario. The experimental scenario was set as an outdoor rectangular environment. The size of the rectangle is 5.5 m ×4.1 m and is composed of four ultra wide band (UWB) anchors laid on the base station, which are located at the same horizontal position. One of the anchors is used as the origin of the rectangle. Two fire scenarios are simulated in the experiment. Scenario 1 is a scenario without an obstacle, whereas Scenario 2 is a scenario with an obstacle.

4.1.2. Description of experimental settings.

Extinguish fire task. In this experiment, M is set as the set of fire situation. m_i is a fire expressed as $m_i = \{\text{num, mp}\}$, where num is the serial number of m_i, which is denoted as num_i. mp is the fire position coordinate of m_i and is denoted as mp_i. In the experiment, the number of fires is set to 3. Their coordinates are $mp_1 = (0.511, 3.31)$, $mp_2 = (3.54, 3.92)$, and $mp_3 = (2.311, 0.889)$.

Fire equipment. The fire equipment a_j has various properties

$$a_j = \{id, p, p\_lst, p\_v, a\_v, r, s, tp, v, \omega, d, fs\}. \quad (4)$$

Table 3 summarises the properties of the fire equipment.

In this experiment, the number of fire equipment items is set to 2: a_1 and a_2. They are randomly placed in this scenario with the real positions being $(4.9, 2.9)$ and $(5.0, 0.1)$, respectively.

Strategy and norm. Several strategies for the fire extinguishing tasks are designed. This study considers different aspects of these strategies, such as the surroundings of the fire and the attributes of each piece of fire equipment. Table 4 lists the norm bases for the FERD system. Norm r_1 creates two instances of fire equipment. Norms r_2 and r_3 are the norms that set the 'control mode' and order the fire extinguishing tasks. Norms r_4∼r_8 are the norms for the 'following strategy'. Norms r_4 and r_5 are the norms for judging 'following strategy' of the fire equipment a_i. Norm r_4 affects the longitudinal and lateral control of the whole fire equipment on the FERD system by the spacing between adjacent fire equipment.

In the considered rectangular area, it is recommended that the linear velocity range of the fire equipment be set at 0.05∼0.5 m/s. The angular velocity range of the fire equipment should be set at 0∼36 rad/s. If fire equipment a_i is found to be farther away from the fire equipment in front, its speed is increased to reduce the distance from the fire equipment in front, and vice versa. If there is a need to create a gap between the fire equipment, norm r_5 is used to instruct the fire equipment to brake urgently in the case of an expected accident. The fire equipment a_i corresponds to the parameters of the 'following' strategy set in norms r_7 and r_8. Norms r_9∼r_{13} are the norms of the 'mobilization' strategy, including 'formation', 'leave

Table 3. Symbol notation.

Symbol	Unit	Description
p_j	(m,m,rad)	Position
p_lst_j	(m,m,rad)	Position list
tp_j	(m,m)	Position of target point
v_j	m/s	Linear velocity of fire equipment
ω_j	rad/s	Angular velocity of fire equipment
d_j	m	Euclidean distance between tp_j and p_j
p_v_j	–	Previous fire equipment's ID
a_v_j	–	Next fire equipment's ID
r_j	–	Role (leader or follower)
s_j	–	State (stuck or normal)
fs_j	–	Following mode (goal following or trajectory following)

team', and 'return' to 'team'. Norms r_{12} and r_{13} are set to correspond to the parameters of the leader or follower.

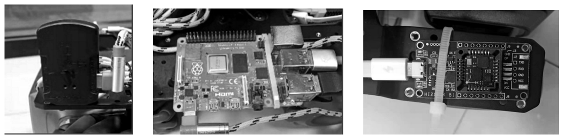
Objective and constraint. The objectives and constraints of the FERD system are expressed as follows:

$$\begin{cases} \min t_{\max}, \\ M = \emptyset, \end{cases} \tag{5}$$

$$s.t.\ p\_lst_j.first = p\_lst_j.last \tag{6}$$

Equation (5) minimizes the maximum time to complete this task and ensures that all fires in M are completely extinguished. Equation (6) establishes that each piece of fire equipment can return to the starting point after completing the task, where $p\_lst_j.first$ denotes the first position of p_lst_j and $p\_lst_j.last$ represents the final position of p_lst_j.

Fig. 5. Fire equipment.



(a) anchor (b) RPI (c) IMU module

Fig. 6. Components of fire equipment.

4.1.3. Experiment platform and equipment. All experiments require a PC platform and an ARM-embedded platform. The application simulation is realized and executed using GBBopen. The ARM-embedded platform used in this experiment is the Raspberry Pi (RPI). RoboMaster (Fig. 5) is used as the fire equipment for the FERD system to perform tasks. The three pieces of fire equipment carried, including the UWB anchor (Fig. 6(a)), the module of IMU (Fig. 6(c)), and the RPI (Fig. 6(b)), are shown in Fig. 6. The Python environment (3.7.0), KIS-CORBA, and ROS platforms are installed into the RPI. The Visual Studio environment (VS2017), KIS-CORBA, Lisp environment (Allegro Common Lisp 10.1 express), LinkTrack technology (NoopLoop for NAssistant applications), and GBBopen are installed on the PC.

4.1.4. Experimental contents. This experiment involves multiple fire equipment set performing fire extinguishing tasks in two scenarios: an outdoor scenario without an obstacle and an outdoor scenario with an obstacle. Multiple pieces of fire equipment are randomly placed at different positions. Their true positions are $(4.9, 2.9)$ and $(5.0, 0.1)$, respectively. The coordinates of the fire extinguishing tasks are $(0.511, 3.31)$, $(3.54, 3.92)$, $(2.311, 0.889)$, respectively. All fire equipment sets share all the norms all of which must be abided by. Each piece of fire equipment can freely switch between appropriate norms based on its state using the sensors it carries. If a fire is successfully extinguished, all fire equipment pieces move to the next fire. When all the fires are extinguished, it is indicated that the fire extinguishing tasks are over. In addition, to check whether multiple fire equipment pieces can automatically switch strategies according to different environments, this experiment sets the control mode of the FERD system to the 'automatic mode'.

4.1.5. Experimental metrics. This study proposes metrics that quantify the effective switching strategy of the FERD system. The evaluation criteria proposed herein can be viewed from two perspectives, being considered from the perspective of the task and from the perspective of the BBS. From the task perspective, the time required to complete the tasks and the length of each fire equipment

Table 4. Norms for the FERD system (γ is a variable, α, β, and λ are thresholds, $\alpha = 0.5$ m, $\beta = 0.6$ m, $\lambda = 2$ m).

Strategy	Norm	Trigger	Action	Expectation
–	r_1	CS start	Create two instances of fire equipment	Two instances of fire equipment
	r_2	Two instances of fire equipment	Set control mode	Control mode is set
	r_3	Two instances of fire equipment	Sort extinguish fire tasks	Extinguish fire tasks are sort
Following strategy	r_4	$d_i > \alpha$	Set fs_i	$fs_i =$ goal following
	r_5	$d_i \leq \alpha$	Set fs_i	$fs_i =$ trajectory following
	r_6	fs_i is set	Set $fs_{a\_v_i}$	$fs_{a\_v_i} = fs_i$
	r_7	$fs_i =$ goal following	Set parameters of a_i	Parameters of a_i is set
	r_8	$fs_i =$ trajectory following	Set parameters of a_i	Parameters of a_i is set
Mobilization strategy	r_9	Extinguish fire tasks are sorted	Fire equipment formation	Leader and follower is determined
	r_{10}	Number of mutated norm $a_i \leq 100$	a_i leave the team	a_i left the team
	r_{11}	Obstacles were cleared in front of a_i	a_i return to team	a_i returned to team
	r_{12}	Leader is determined	Set leader related parameters	Leader related parameter is set
	r_{13}	Followers are determined	Set follower related parameter	Follower related parameters is set
Motor strategy	r_{14}	$d_i > \beta$	$\gamma \leftarrow d_i$ then a_i forward	$d_i < \gamma$
	r_{15}	$d_i \leq \beta$	$\gamma \leftarrow d_i$ then a_i stop	$d_i < \gamma$
	r_{16}	a_i and a_j collide	a_i fallback	a_i and a_j no collide
Obstacle avoidance strategy	r_{17}	$D_{a_i a_{a\_p_i}} > \lambda$	$a_{a\_p_i}$ virtual circle	Attributes of $a_{a\_p_i}$ are changed
	r_{18}	$D_{a_i a_{a\_p_i}} = \lambda$	$a_{a\_p_i}$ virtual isosceles triangle	Attributes of $a_{a\_p_i}$ are changed
	r_{19}	$D_{a_i a_{a\_p_i}} < \lambda$	$a_{a\_p_i}$ virtual right triangle	Attributes of $a_{a\_p_i}$ are changed
Evolutionary strategy	r_{20}	$d_i \geq \gamma$	Alarm	$s_i =$ stuck
	r_{21}	$d_i \geq \gamma$	Norm evolution of crossover	Evolved norms
	r_{22}	$d_i \geq \gamma$	Norm production of other methods	Generated norms

path serve as the evaluation criteria. From the perspective of the BBS, the number of switches between different strategies per piece of fire equipment is the evaluation criterion.

4.2. Experimental steps.

4.2.1. Cross-language data sharing. Because fire equipment is written in Python, and GBBopen is written in Lisp, cross-language communication is required. These experiments focused on the communication between Python and Lisp and finally applied the FERD system. The steps in cross-language communication are summarised as follows.

1. **Describe interfaces by the interface description language.** This experiment involves data sharing between Python and Lisp. These two interfaces are defined with the interface description language (IDL). One of the IDL interfaces is defined using Python to transfer the data to Lisp. The other IDL interface is defined using Lisp and transmits data to Python. Subsequently, the IDL files are compiled. After IDL is compiled, Python generates two files: '_.py' and '_skel.py', while Lisp generates three files: '-sysdcl.lisp', '-procs.lisp', and '-basics.lisp'.

2. **Server and client in KIS-CORBA.** The overall framework of the KIS-CORBA server program is generated using an IDL compiler, and the client and server are established. It is noteworthy that the IDL used by the server and client must be consistent. Otherwise, cross-language data sharing is not normal communication. Finally, the data of the fire equipment is sent to GBBopen, including pose, velocity, distance, etc. GBBopen sends the data to the fire equipment such as target point, speed, etc.

4.2.2. APIs of fire equipment. DJI-Innovations (DJI) provides several API instances[2] written in Python. This experiment is conducted to realize communication between the fire equipment and GBBopen and to control the wheels of the fire equipment. For example, an API program is added to control the wheels in GBBopen.

4.2.3. Pose information of fire equipment. This experiment assumes that the position information obtained by multiple sensors indicates the position information of the actual fire equipment. The remote server is connected to control the fire equipment through a secure shell (SSH). Positioning errors occur due to various factors (e.g., the deployment mode of the base station and the weakening of the signal) in the external environment. The average value of the real-time position of each anchor is calculated to reduce noise interference. In addition, the three-axis attitude angle is obtained using the IMU technology. The ROS serial port then reads the obtained position data and publishes them in the form of topics. Finally, they are transmitted to GBBopen through KIS-CORBA.

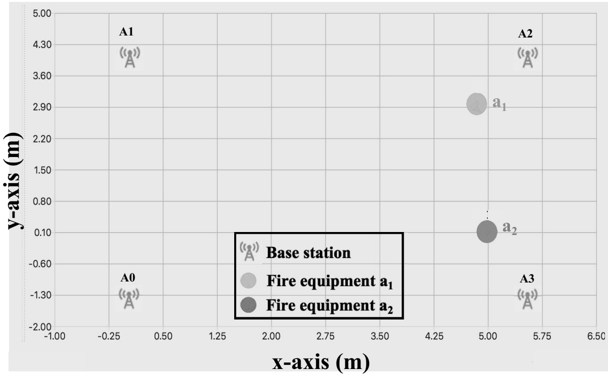


Fig. 7. Two fire equipment settings in the experimental scenario.

Figure 7 displays the positions of the carried anchor in two fire equipment settings, where A_0, A_1, A_2, and A_3 are four manually arranged base stations. Additionally, each piece of the fire equipment is mapped to a point. The position of a_1 is displayed in light gray, whereas the position of a_2 is displayed in gray. According to the average value of the calculated posture, p_1 of fire equipment a_1 is $(4.929, 3.049, 1.2)$, and p_2 of fire equipment a_2 is $(4.928, 0.065, 3.8)$.

4.2.4. Fire equipment performing tasks. Because GBBopen is developed in the Lisp programming language, the server and client are embedded in the GBBopen program. Each KS in GBBopen can read or send data using KIS-CORBA. The detailed processes for controlling the fire equipment using GBBopen are described below. First, GBBopen sends the target position using KIS-CORBA to each piece of fire equipment. The position and other information of the two pieces of fire equipment are obtained in real time using sensors. Each piece of fire equipment can adjust its own linear and angular velocities based on the target position obtained from GBBopen. The pose, distance, and velocity of the fire equipment are sent to GBBopen using KIS-CORBA. The CS then switches between these KSs in different scenarios. The actions of the switched KS cause the position of each piece of fire equipment to be adjusted such that each piece of fire equipment takes the fastest path to the target point. Finally, the process ends when the fire extinguishing tasks are completed.

[2]https://robomaster-dev.readthedocs.io/zh_CN/latest/.

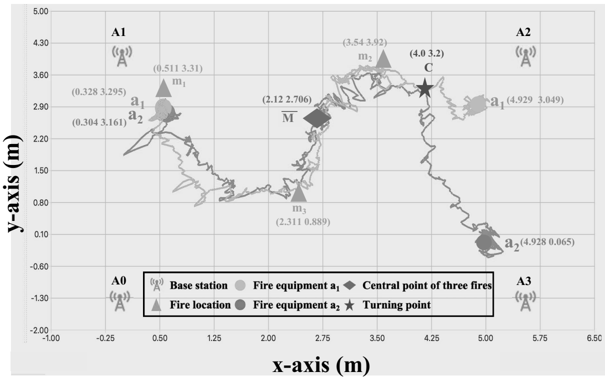
Notably, there is a communication delay during system operation such that Tri_i of norm r_i is continuously satisfied. This phenomenon is known as the 'Zeno effect.' Our solution is to add a global variable to the FERD system; a judgment of the global variable value should also be added to Tri_i. When Tri_i of norm r_i is satisfied, the value of the global variable is immediately changed to prevent the infinite triggering of norm r_i in a finite time. For example, a global variable 'tag', tag $= 1$. Tri_a of norm r_a is added to the judgment expression for the variable. Then, Tri_a = $\{D > 1$ m, $power > 20\%$, tag $= 1\}$. When Tri_a is satisfied, but act_a is not executed. The value of the tag is changed immediately. At this point, Tri_a is not satisfied, and r_a is not triggered again in finite time. When Act_i is executed, the value of tag is restored to 1. In this case, Tri_a can be rejudged.

4.3. Results and analysis.

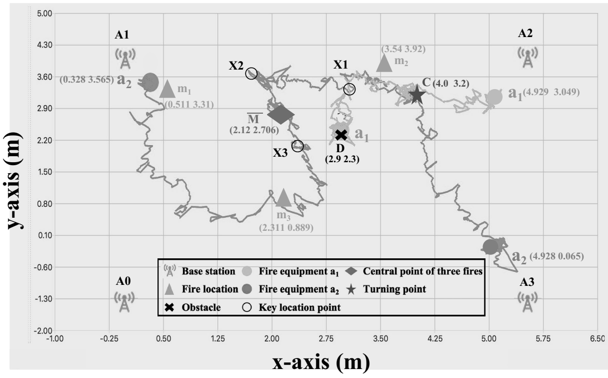
4.3.1. Experimental results. Some experimental results for the fire equipment in the two scenarios have been obtained. Figure 8 shows the trajectories of two pieces of fire equipment. A_0, A_1, A_2 and A_3 represent four manually arranged base stations, and mp_1, mp_2, and mp_3 represent the fire positions. $\bar{M}$ represents the centre point of all fire positions, and is calculated as the average value of the x-axis and y-axis using the coordinates of all fire locations. The calculation of $\bar{M}$ results in (2.120,2.706). The light gray and gray lines represent the tracks of the fire equipment a_1 and a_2, respectively, from right to left. They extinguish fire tasks in the order of m_2, m_3, and m_1.

Figure 8 demonstrates that the distance from a_1 to $\bar{M}$ is smaller than that from a_2 to $\bar{M}$. According to norms r_2 and r_3, the FERD system switches to the 'formation' strategy, where a_1 is the leader and a_2 is the follower. In addition, we find that the distance from a_1 to m_2, m_3, and m_1 increases in turn. Therefore, the order of the two pieces of fire equipment used to extinguish fire tasks is m_2, m_3, and m_1. The distance between a_1 and m_2 is compared with the threshold β, and the distance between a_1 and a_2 is compared with threshold β. According to norm r_{14}~r_{16}, fire equipment a_1 and a_2 can independently be switched to 'forward', 'stop' and 'fallback' in the motor strategy.

Figure 8(a) shows the experimental results of Scenario 1. It is obvious from Fig. 8(a) that the turning point C is the switching point from 'goal following' to 'trajectory following.' It can be seen from this figure that a_2 follows a_1 as a follower when it moves towards m_2. Before turning point C, the trends for the red and green tracks are observed to be significantly different. Because a_1 has not reached turning point C, the distance between a_1 and m_2 is greater than threshold α. According to norms



(a) for the outdoor scenario without an obstacle



(b) for the outdoor scenario with an obstacle

Fig. 8. Trajectories of all fire equipment pieces in different scenarios.

r_4, r_6, and r_8, the following strategy of a_2 is switched to 'goal following' in the process of moving to a_1. After the turning point C, the motion trajectories of a_1 and a_2 are consistent. In this experiment, the distance between position a_2 and position a_1 is set to be less than 0.3 m, which implies that the positions of a_2 and a_1 coincide. The distance from a_1 to m_2 is equal to the threshold α. According to norms r_5 and r_8, the following strategy of a_2 is switched to 'trajectory following' in the process of moving to a_1. Finally, a_1 and a_2 start from turning point C and jointly complete the fire extinguishing tasks for m_2, m_3, and m_1.

Figure 8(b) shows the experimental results of Scenario 2. After a_1 and a_2 extinguish the fire m_2, they continue to extinguish fire m_3. Subsequently, a_2 continues to follow a_1 along the 'trajectory following.' When a_1 drives towards m_3, it passes through X_1 and becomes stuck by obstacle D. According to norm r_{10}, fire equipment a_1 automatically switches strategy to 'leave team'. In addition, the 'crossover' is automatically switched to evolve the original norm r_{14}. Norm r_{14} is detected as a mutated norm, leading to the exception of other norms such as norm r_{20}. The Exp_{20} of norm r_{20} is denoted as Exp_{20} $= \{s_i$ is stuck$\}$. In fact, s_1 is normal; therefore, r_{20} is a mutated norm. Then, Tri_{14} of the norm r_{14} evolves to $Tri_{14} = \{d_i > \beta,\ s_i = \text{normal}\}$. Because s_1 does not satisfy Tri_4 of norm r_4, fire equipment a_1 cannot be moved. For fire equipment a_1, v_2 is relatively small, and a_2 does not reach obstacle D. Therefore, according to norm r_{19}, fire equipment a_2 switches to the 'virtual right triangle' and changes its role of the leader. The track of fire equipment a_2 passes through $X_1 \rightarrow X_2 \rightarrow X_3$. It can be said that fire equipment a_2 successfully bypasses the obstacles and reached the fire m_3; this indicates that in the case of an obstacle, the FERD system can still switch various strategies to effectively complete the fire extinguishing tasks. Therefore, the switchability, agility, and robustness of the FERD system are verified using the two scenarios shown in Fig. 8.

4.3.2. Evaluation. In the experiment, different scenarios required different times to complete the fire extinguishing tasks. The results of the FERD system performing the fire suppression tasks were evaluated using the evaluation metrics in Section 4.1.5. The time to complete the fire extinguishing tasks, the length of each fire equipment path, and the number of switching strategies were counted in the two scenarios. Table 5 lists the completion times and lengths of each fire equipment path. Table 6 shows the change when switching between different strategies for the two fire equipment pieces.

Table 5 provides the times to complete the fire extinguishing tasks and the lengths of each fire equipment path in scenarios 1 and 2. In Scenario 1, the time required by the FERD system to complete the fire-extinguishing tasks was 4 min 51 s. In Scenario 2, the time taken by the FERD system to complete the fire extinguishing tasks was 7 min 37 s, which was 2 min 56 s longer than the time used in Scenario 1. In Scenario 1, the length of the path of a_1 was 10.52 m, and the length of the path of a_2 was 12.86 m. In Scenario 2, the length of the path of a_1 was 5.01 m, and the length of the path of a_2 was 14.13 m.

As shown in Table 6, there is a little difference in the number of switching strategies between the two pieces of fire equipment in different scenarios. In Scenario 1, the total number of switching strategies for fire equipment a_1, a_2 were 648 and 745, respectively. However, in scenario 2, the total number of switching strategies for fire equipment a_1 was 448, which was 30.8% less than that in Scenario 1. The total number of switching strategies for fire equipment a_2 was 937, which was 25.8% more than in Scenario 1. Note that there were no obstacles in scenario 1. The number of switching strategies for fire equipment that a_1 switched to 'forward' is 534, and the number of switching strategies for fire equipment that a_2 switched to 'forward' is 646. However, there were obstacles in Scenario 2. In Scenario 2, the number of switching strategies for fire equipment a_1 was 322, which was 39.7% fewer than that in Scenario 1. The number

Table 5. Completion time and length of each piece of the fire equipment path.

Scenario	Scenario 1		Scenario 2	
Completion time	4 min 51 s		7 min 37 s	
Fire equipment	a_1	a_2	a_1	a_2
Length of the path (m)	10.52	12.86	5.01	14.13

Table 6. Strategies and the number of switches for each piece of fire equipment (times).

Scenario	Scenario 1		Scenario 2	
Fire equipment	a_1	a_2	a_1	a_2
Formation	1	1	1	1
Trajectory following	27	1	35	1
Goal following	75	76	81	82
Forward	534	646	322	816
Stop	5	17	4	27
Fallback	6	4	4	9
Leave team	-	-	1	0
Virtual right triangle	-	-	0	1
Total	648	745	448	937
	1393		1385	

of switching strategies for fire equipment a_2 was 816, which is 26.3% greater than that in Scenario 1. This is because fire equipment a_1 was intercepted by this obstacle and was forced to stop and the system autonomously switched to the 'evolutionary strategy.' Finally, norm r_{14} evolved; namely, the judgment expression of Tri_{14} was increased. The fire equipment s_1 was not satisfied the trigger condition of r_{14}, so a_1 was switched to 'forward'. In addition, the fire equipment a_2 makes the driving path longer than that in scenario 1 because a_2 is autonomously switched to the 'virtual right triangle' approach. Based on this analysis, we conclude that the FERD system can autonomously switch strategies in different scenarios.

5. Conclusion

In practical applications, it is of great research significance that agents can switch strategies autonomously, effectively, and quickly. The existing research switching strategies mainly focus on reactions to specific scenarios, and very little attention has been paid to the effects of the task level. This paper proposes a framework of ASTS, which is a framework that can automatically switch strategies according to the different scenarios to reaction to various emergencies. To prove the autonomy and switchability of the framework, this paper has developed a FERD system applied to fire scenarios. It aims to switch between different strategies and control modes for multiple types of fire equipment based on the on-site environment. It improves the adaptability and flexibility of multiple agents in an open environment and represents a significant step toward switching strategies at the task level.

In future work, the proposed FERD system will be implemented on other platforms (indoors) for verification. It is also considered to address more complex situations, such as external environments (e.g., roads or intersections) and participants (e.g., pedestrians), which can add multiple strategies and control modes. Moreover, the strategies and control modes formulated in the FERD system are highly dependent on network communication and sensors (e.g., positioning) carried fire equipment. The existing perception and real-time problems (e.g., positioning by a sensor and network latency) are potential problems of the autonomous switching strategies of multiple fire equipment, which require further research. For the perception problem of each piece of fire equipment, it is necessary to use multiple sensors to assist in correcting the perception and further reduce sensor errors.

Acknowledgment

Xianchang Wang and Bingyu Lv have made similar contributions to this article and are designated as co-first authors, Kaiyu Wang is the second author and Rui Zhang is the third one.

References

Bai, S., Jia, X., Cheng, Z. and Guo, B. (2021). Operation strategy optimization for on-orbit satellite subsystems considering multiple active switching, *Reliability Engineering and System Safety* **215**: 107765, DOI: 10.1016/j.ress.2021.107765.

Hook, J., El-Sedky, S., Silva, V.D. and Kondoz, A.M. (2021). Learning data-driven decision-making policies in multi-agent environments for autonomous systems, *Cognitive Systems Research* **65**(2): 40–49, DOI: 10.1016/j.cogsys.2020.09.006.

Kanazawa, A., Kinugawa, J., and Kosuge, K. (2021). Motion planning for human-robot collaboration using an objective-switching strategy, *IEEE Transactions on Human-Machine Systems* **51**(6): 590–600, DOI: 10.1109/THMS.2021.3112953.

Landgren, P., Srivastava, V. and Leonard, N.E. (2021). Distributed cooperative decision making in multi-agent multi-armed bandits, *Automatica* **125**: 109445, DOI: 10.1016/j.automatica.2020.109445.

Li, Y., Ma, D., An, Z., Wang, Z., Zhong, Y., Chen, S. and Feng, C. (2022). V2x-sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving, *IEEE Robotics and Automation Letters* **7**(4): 10914–10921, DOI: 10.1109/LRA.2022.3192802.

Lin, J., Su, W., Xiao, L. and Jiang, X. (2018). Adaptive modulation switching strategy based on Q-learning for underwater acoustic communication channel, *International Conference on Underwater Networks & Systems, WUWNet 2018, Shenzhen, China*, pp. 38:1–38:5.

Liu, G., Wu, S., Zhu, L., Wang, J. and Lv, Q. (2022). Fast and smooth trajectory planning for a class of linear systems based on parameter and constraint reduction, *International Journal of Applied Mathematics and Computer Science* **32**(1): 11–21, DOI: 10.34768/amcs-2022-0002.

Ma, J., Cheng, Z., Zhang, X., Mamun, A.A., de Silva, C.W. and Lee, T.H. (2020). Data-driven predictive control for multi-agent decision making with chance constraints, *arXiv*: abs/2011.03213.

Ming, Y., Chen, J., Dong, Y. and Wang, Z. (2022). Evolutionary game based strategy selection for hybrid v2v communications, *IEEE Transactions on Vehicular Technology* **71**(2): 2128–2133, DOI: 10.1109/TVT.2021.3132025.

Nimmolrat, A., Sutham, K. and Thinnukool, O. (2021). Patient triage system for supporting the operation of dispatch centres and rescue teams, *BMC Medical Informatics and Decision Making* **21**(1): 68–84, DOI: 10.1186/s12911-021-01440-x.

Oguz-Ekim, P., Bostanci, B., Tekkk, S.. and Synmez, E. (2020). The EKF based localization and initialization algorithms with UWB and odometry for indoor applications and ROS ecosystem, *International Conference on Advanced Computing and Applications, SIU 2020, Gaziantep, Turkey*, pp. 1–4.

Roy, D., Chowdhury, A., Maitra, M. and Bhattacharya, S. (2018). Multi-robot virtual structure switching and formation changing strategy in an unknown occluded environment, *IEEE Conference on Computer Communications, IROS 2018, Madrid, Spain*, pp. 4854–4861.

Sengupta, A. and Yasser Mohammad, S.N. (2021). An autonomous negotiating agent framework with reinforcement learning based strategies and adaptive strategy switching mechanism, *arXiv*: abs/2102.03588.

Shin, M.E., Kang, T. and Kim, S. (2018). Blackboard architecture for detecting and notifying failures for component-based unmanned systems, *Journal of Intelligent and Robotic Systems* **90**(2): 571–585, DOI: 10.1007/s10846-017-0677-4.

Sun, K. and Liu, X. (2021). Path planning for an autonomous underwater vehicle in a cluttered underwater environment based on the heat method, *International Journal of Applied Mathematics and Computer Science* **31**(2): 289–301, DOI: 10.34768/amcs-2021-0020.

Tan, J., Khalili, R., Karl, H. and Hecker, A. (2022). Multi-agent distributed reinforcement learning for making decentralized offloading decisions, *IEEE Conference on Computer Communications, INFOCOM 2022, London, UK*, pp. 2098–2107.

Wang, H., Yan, J., Han, S. and Liu, Y. (2020a). Switching strategy of the low wind speed wind turbine based on real-time wind process prediction for the integration of wind power and EVS, *Renewable Energy* **157**: 256–272, DOI: 10.1016/j.renene.2020.04.132.

Wang, P., Yang, J., Jin, Y. and Wang, J. (2020b). Research on allocation and dispatching strategies of rescue vehicles in emergency situation on the freeway, *International Conference on Control, Automation, Robotics and Vision, ICARCV 2020, Shenzhen, China*, pp. 130–135.

Wang, X., Fu, R. and Zhang, R. (2020c). MONE: Mutation oriented norm evolution, *IEEE Access* **8**: 205386–205395, DOI: 10.1109/ACCESS.2020.3037798.

Wu, P., Chu, F., Che, A. and Zhou, M. (2018). Bi-objective scheduling of fire engines for fighting forest fires: New optimization approaches, *IEEE Transactions on Intelligent Transportation Systems* **19**(4): 1140–1151, DOI: 10.1109/TITS.2017.2717188.

Zhao, T., Zhang, W., Zhao, H. and Jin, Z. (2017). A reinforcement learning-based framework for the generation and evolution of adaptation rules, *International Conference on Autonomic Computing, ICAC 2017, Columbus, USA*, pp. 103–112.

Zhou, Z. (2021). *Large-Scale Multi-Agent Decision-Making Using Mean Field Game Theory and Reinforcement Learning*, PhD thesis, University of Nevada, Reno.

Bingyu Lv received her BS degree in management from the Heilongjiang Institute of Technology in 2015 and her MS degree from the School of Petroleum Engineering of Northeast Petroleum University in 2019. She is currently pursuing her PhD degree with the Department of Computer Science and Technology, Jilin University, China. Her present research interests include artificial intelligence and multi-agent systems.

Xianchang Wang received his PhD degree in computer science and engineering from the National University of Defense Technology, China, in 1991. He is currently a professor with the College of Computer Science and Technology, Jilin University, China. His research interests include artificial intelligence and multi-agent systems.

Kaiyu Wang received his BS degree in computer science and technology from Jilin University, China, in 2019. He is currently pursuing his PhD degree with the College of Computer Science and Technology, Jilin University. His present research interests include multi-agent reinforcement learning and robust reinforcement learning.

Rui Zhang holds a PhD degree in informatics from the University of Trento, Italy. He is currently an associate professor with Jilin University, China. He is the author of the book *Relation-Based Access Control* (IOS Press, 2010). His research interests include knowledge representation and data integration.

Received: 14 December 2022
Revised: 19 February 2023
Re-revised: 18 April 2023
Accepted: 20 April 2023