An Immunity-Based Distributed Multiagent-Control Framework

Henry Y. K. Lau, *Member, IEEE*, and Vicky W. K. Wong

***Abstract*—The human immune system is a complex system of cells, molecules, tissues, and diverse organs that can provide us with primary defense against pathogenic organisms. These components are highly interactive and execute the immune response in a coordinated and specific manner. This paper presents a formal mathematical model of an artificial immune system (AIS)-based control framework. The framework aims to provide an integrated solution to control and coordinate complex distributed systems with a large number of autonomous agents such as automated warehouses, distribution centers, and automated material-handling systems. The control framework consists of a set of AIS agents working in response to the changing environment and the occurrence of tasks. The AIS agents manipulate their capabilities to derive appropriate responses to tackle different problems. A methodology describing the response-manipulation algorithm of the AIS agents and their ability to generate new capabilities is discussed in this paper. Through response manipulation and knowledge building, a self-organized and fully distributed system with agents that are able to adapt and accommodate in a dynamic environment via distributed decision making and interagent communication is achieved.**

***Index Terms*—Artificial immune system (AIS), complex distributed control, multiagent system.**

I. Introduction

THE HUMAN immune system performs pattern recognition, learning, and retains memory of the antigens that it has fought [1]. It is a highly distributed complex system with a collection of independently operating agents. These agents communicate via chemical signals, cell-to-cell contact, and secretion of molecules. The immune system is a self-organized and fully distributed multiagent system with properties of specificity, diversity, memory management, self-organization, and adaptive control. The properties of distributive control and self-organization impart a high degree of robustness that has created great interest in adopting the immune-system paradigm to various engineering systems; and this engineering analog is called artificial immune system (AIS).

AIS has been studied widely in the fields of artificial intelligence (AI) due to its deep inspiration to engineering sciences. The essences of human immune system are imitated to perform complicated tasks, such as adaptive control, learning tactics, communication strategies, and memory managements. These special properties have been integrated and adopted in various complex systems for problem solving. They include parallel searching [2], detection systems [3], [4], autonomous agents [5], [6], computing systems [7], [8], fault-tolerance systems [9], [10], and decentralized mechatronic control [11].

In [12] and [13], we proposed a control framework that has the ability to manage, coordinate, and schedule a fleet of agents by imitating the human immunity mechanism. The AIS-based control framework operates various kinds of multiagent systems. It addresses how agents of different intelligence levels achieve tasks in a dynamic environment through communication and strategic behavioral control. Previous studies have shown that an AIS-based control framework is more efficient in operating a fleet of autonomous guided vehicles (AGVs) in an automated warehouse than the traditional centralized control [12]. This paper focuses on the development of strategic control methods with AIS for these agents. AIS agents are employed with full autonomy in decision making and communication. Their internal behaviors provide mutual understanding among them and allow them to cooperate strategically.

To quantify individual and group behaviors of AIS agents, a formal mathematical model is presented in this paper. Being a multiagent-control framework, cooperation is indispensable. A cooperation model is presented with the aid of a behavior study. One significant mechanism of the immune system is its ability to fight against antigens with different immune responses. Hence, the immune metaphor allows AIS agents to operate autonomously in a dynamic environment by capability manipulation where different responses are exhibited. The capability-manipulation methodology is described in detail in this paper. Through capability manipulation, AIS agents are able to determine appropriate responses towards different problems and situations. Hence, agents under this AIS-based control are fully autonomous and adaptive.

This paper proceeds as follows: Section II starts by giving a brief overview of the human immune system. Section III reviews our proposed AIS-based multiagent control. Section IV presents a formal mathematical description of the AIS-based control framework and Section V presents a cooperation model in terms of communication and interaction between AIS agents. Section VI presents the methodology of capability manipulation of the AIS agents with an illustrative example to demonstrate how to generate new capabilities. Section VII evaluates the overall operational scheme of our control framework and presents the results of a simulation study.

Manuscript received December 1, 2004; revised April 28, 2005. This paper was supported in part by the Research Grant Council of the Hong Kong Special Administrative Region under Competitive Earmarked Research Grant (CERG) Project HKU7079/02E. This paper was recommended by Associate Editor M. Zhou.

The authors are with the Department of Industrial and Manufacturing Systems Engineering, The University of Hong Kong, Hong Kong (e-mail: hyklau@hku.hk; vickywong@hkusua.hku.hk).

Digital Object Identifier 10.1109/TSMCA.2006.859103

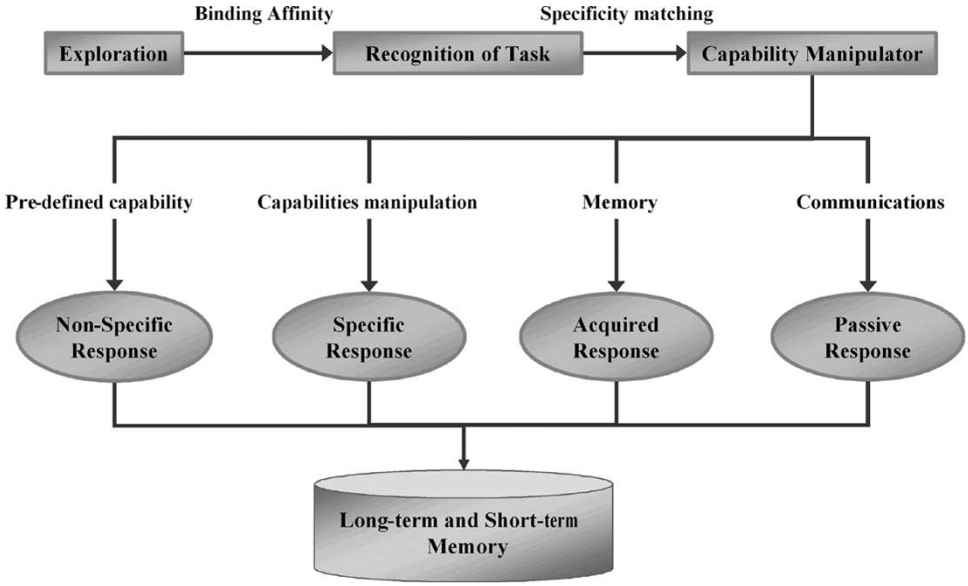


Fig. 1. Architecture of the control framework for the individual AIS agent.

II. An Overview of Human Immune System

The human immune system is a complex system consisting of diverse organs, tissues, innate cells, and acquired cells distributed throughout our body. These components are interrelated, and act in a highly coordinated and specific manner. Interactions between these components generate immune responses of either innate or acquired immunity. Immunity performs a number of vital functions, for example, the elimination of invading antigens and the activation of amplification mechanisms in developing protective antibodies [14].

Innate immunity is inborn and unchanging. It provides resistance to a variety of antigens during their first exposure to a human body. This general defense operates nonspecifically during the early phase of an immune response and is known as primary immune response. On the other hand, acquired immunity develops during the lifetime of a person and is based partly on the person's experiences. This immunity is antigen specific and is activated during the first exposure to antigens. The antibodies and immune cells eliminate the antigens following activation. Some of the immune cells become memory cells after the elimination and on subsequent reencounter of the same antigen; the immune system is primed to destroy these antigens more quickly [15]. This stronger and faster immune response is known as secondary immune response.

To generate successful immune responses, immune cells play a major role in destroying foreign antigens. Lymphocytes are the main antigen killer among all the immune cells. They have special binding areas, known as receptors, which can structurally determine and react with specific foreign antigens. The specificity of these cells is developed before the antigen is introduced. An antigen does not induce the appearance of immune cells; rather, it selects and stimulates preexisting antigen-specific cells by interacting with their receptors [15]. The group of immune cells specific to an antigen is called a clone of cells. Diversity and specificity are a result of having millions of different clones that demonstrate unique antigen specificity.

III. AIS-Based Multiagent-Control Framework

Specificity [16], inducibility, diversity [2], memory [17], [18], distinguishing self from nonself [19], [20], and self-regulation are the six major characteristics of the human immune system. These characteristics enable the human immune system to explore very high dimensions in solving engineering problems. Various approaches have adopted the properties of the immune system in the implementation of multiagent control. The cooperative controls in [5] and [21] use group behavior mechanism of the immune system for autonomous mobile robots. A distributed autonomous robotic system (DARS) that utilizes the AIS techniques shows a noticeable improvement in [22]. Autonomous navigation is developed in [23] to investigate an autonomous control system of mobile robots, which is based on the immune-network theory. Since AIS has founded its popularity in developing various kinds of multiagent systems, a control framework based on this biological metaphor is proposed to operate a fleet of AGVs for automated material handling with an architecture (Fig. 1) that defines the AIS-based control strategies of an individual AIS agent [12].

The control framework organizes a fleet of agents in a dynamic environment. It is developed based on the fully distributed mechanism of biological immunity. The immune system is a special type of multiagent system where each agent or immune element has specific behavior patterns and functions for a particular antigen. The behavior of the AIS agents of the control framework depends on the environment as well as individual behavior. An AIS agent has a specific set of capabilities that determine their fundamental intelligence. This intelligence can be enhanced through interagent communication or via explorations in the environment. A behavioral-based distributed

network is adopted for controlling AIS agents. This is a distributed network based on the transition of agents' behavior state. The operations of AIS agents are not predefined in the planning stage but are altered dynamically to adapt to the corresponding working environment. This distributed and nondeterministic control inherits the following characteristics and functions that are offered by the human immune system.

1) Self-organization: AIS agents having the unique capability to determine responses to achieve common tasks through independent decision making and communications.
2) Self-/Non-self-recognition: AIS agents identify tasks during random exploration. Affinity function, an index for non-self-identification, is used to evaluate the stimulation of a specific task for an agent. This non-self-recognition depends solely on the stimulation provided by a task, which agents do not have any prior information on, such as location and complexity of the task. Hence, with a simple set of capabilities predefined for each AIS agent, the AIS agents are extremely flexible in addressing different types of problems.
3) Adaptability: AIS agents adjust their behaviors when tackling a problem. They manipulate the best response to solve the problem by evaluating the specificity of its capability. New knowledge or functionality is gained when coping with variations of tasks and environment.
4) Robustness: Failure of any agent to execute an operation will not cripple the overall system. Since the control framework is aimed to be fully decentralized, no fixed dependence is assigned between tasks and agents.
5) Diversity: AIS agents are able to learn from experiences. These experiences allow the agents to manipulate their capabilities and enhance their fundamental intelligence in solving different problems.

The control framework provides a set of rules that guides and determines the behavior of individual AIS agent in response to the changing environment. Through the manipulations of the rules, unknown events and dynamic variations of the workplace can be investigated effectively. The control framework depicted in Fig. 1 demonstrates how AIS agents can recognize the task independently, tackle the task with specificity, manipulate new capabilities spontaneously, and cooperate with each other effectively. Each of the essential components given in Fig. 1 is discussed in the latter part of this paper. A formal mathematical model and cooperation model that specify the operation of AIS agents in the control framework and a detailed analysis of the response-manipulation algorithm that describes how the AIS agents generate appropriate responses and new capability sets in problem solving are described.

Fig. 1 shows the flow of control of an AIS agent under our control framework. Upon deployment in a workplace, AIS agents first explore their surrounding environment within their sensory ranges (SRs). Based on the measure of binding affinity, agents recognize and approach tasks. After a task has been recognized by an agent, it will then manipulate its capabilities by a function called specificity matching. Details of both binding affinity and specificity matching are given in Section IV. The capability manipulator allows agents to perform appropriate responses in tackling tasks with different complexities.

TABLE I
EXAMPLES OF LONG-TERM AND SHORT-TERM MEMORIES

Long-term Memory
Pre-defined capabilities
Newly generated capabilities by specific response
Specificity matching function for encountered tasks
Short-term Memory
Instruction received from passive response
Teammates for tackling a cooperative task
Binding affinity function for sensed tasks

Four types of responses, in relation to the human immune system, have been defined in the control framework. They are nonspecific, specific, acquired, and passive responses. The nonspecific response is a fast and straightforward response to tackle simple and general tasks. Agents execute predefined capability in performing the nonspecific response. For more complicated tasks, agents require the generation of a new set of capabilities that are specific to these complex tasks. The specific response, which allows agents to generate new set of capabilities by rearranging or recombining their fundamental capabilities, is therefore performed. The newly generated capability is then stored in the agent's memory in the form of acquired response for future use. Passive response is solely for cooperative tasks. Agents who are tackling a cooperative task send stimulation to activate their surrounding agents to join in if more agents are required. Activated agents who are not capable of tackling the cooperative task will receive instructions solely from the initiating agent for completing the task. This is a passive response where agents are activated by others to tackle a task instead of recognizing a task by them in the first place. The relationships of the four responses and the detail of the response-manipulation algorithm are discussed in Section VI.

The memory of the AIS agents are divided into long-term and short-term memories. Long-term memory stores information that is essential in completing all the tasks in the workplace. Short-term memory stores data for temporary use. Examples of long-term and short-term memories are listed in Table I.

IV. MATHEMATICAL MODELING

The components of the distributed AIS-based multiagent-control framework shown in Fig. 1 are formally described in this section. As inspired from the mechanisms of biological immunity, each AIS agent has its own behavior. The AIS agents cooperate via communication and sharing of local information in order to achieve common tasks. Agents employed in our control framework are abstracted as an independent agent that carries local information, searches for solution space, and exhibits robust behavior to accomplish different tasks.

The basic attributes of the AIS-based control framework includes a set of agents that is operated in the system, a set of tasks that is located in the workplace, an SR of an agent

that enables it to gain or receive information of its surrounding environment, and a communication range (CR) that allows the agent to exchange information or message to each other.

A is a set of agents indexed by j

$$A = \{a_1, a_2, \ldots, a_j\}, \qquad \text{where } j = 1, 2, \ldots, n. \quad (1)$$

G is a set of tasks indexed by i

$$G = \{g_1, g_2, \ldots, g_i\}, \qquad \text{where } i = 1, 2, \ldots, m. \quad (2)$$

The workplace environment Env is defined as

$$\text{Env} = a_j \cup g_i \qquad \forall i, j. \quad (3)$$

A. Binding Affinity

AIS agents employ the measure of binding affinity (β) in task recognition to identify and approach a targeted task in a dynamic environment. The binding affinity is enumerated by the distance between an agent and a particular task (d_{ij}), task occurrence frequency (f_{ij}), and specificity-matching function (r_{ij}). The definition of binding affinity is given as follows:

$$\beta_{ij} = f(d_{ij}, f_{ij}, r_{ij}) \quad (4)$$

$$\beta_{ij} = w_1(d_{ij})^{-1} + w_2(f_{ij}) + w_3(r_{ij}) \quad (5)$$

where w_1, w_2, and w_3 are the weightings for d_{ij}, f_{ij}, and r_{ij}, respectively.

Distance is an important element in measuring the affinity between an antibody and an antigen in biological immunity. Manhattan distance is chosen as part of the affinity function in our framework to measure the distance between an agent and a task. This is because Manhattan distance is computationally more efficient than Euclidean distance especially in parallel implementation of the algorithms as shown in [24] and [25]. Moreover, Freitas and Timmis [24] demonstrated that Manhattan distance tends to be more robust to noisy data. As Euclidean distance over emphasizes large differences in the values of the coordinates by exponentiation or square root operation, a single error in the value of one coordinate in the agent or task vectors can be considerably amplified by the Euclidean distance. Hence, Manhattan distance is more appropriate in constructing the distance function for our framework due to its advantages in computational efficiency and more importantly, its suitability in the proposed domain.

Hence, $d_{ij(E)}$ is the Manhattan distance between an agent j and a task i in a two-dimensional plane

$$d_{ij(M)} = |x_i - x_j| + |y_i - y_j|. \quad (6)$$

In biological immunity, distance is not the only concern for affinity. The affinity between an antibody and an antigen involves several processes such as pattern recognition, hydrogen binding, and noncovalent and van der Waals interactions. Hence, the binding affinity function of our framework consists of three parameters between an agent and a task. Besides Manhattan distance, the other two parameters are task occurrence frequency and specificity-matching function. Task occurrence frequency enumerates the task-achievement history of every agent. Our framework is aimed for modeling automated warehousing operations. It is common to have the same type of tasks repeated in the workplace. Task occurrence frequency can therefore increase the affinity between an agent and a particular type of task by evaluating how many times the agent has achieved that task successfully in the past. Hence, agents have a tendency towards more familiar tasks.

f_{ij} is the frequency of occurrence of a task, where O_i is the number of occurrences of the task i

$$f_{ij} = \frac{O_i}{\sum_{x=1}^{m} O_x}. \quad (7)$$

A specificity-matching function is a function that follows the concept of pattern recognition in biological immunity. This function verifies the feasibility of an agent to handle a given task. Since an immune cell recognizes an antigen structurally, we transform task complexity into codes so that an agent can match these codes with their capabilities. The closer is the match, the higher is the affinity between the agent and the task. Specificity matching is a crucial part of our framework since it also determines which responses an agent should perform to tackle a given task. A detailed description of agent–task specificity matching is presented in Section VI.

r_{ij} is the specificity-matching function that describes an agent's ability and familiarity with a task

$$r_{ij} = \frac{1}{R_{\mathrm{I}}^S \cdot R_{\mathrm{L}}}. \quad (8)$$

Here, R_{I} is the corresponding index for relative intelligence between an agent and a task, R_{L} is a measure of compatibility of matching between the capability of an agent and the task, and S is the relative score that highlights the efficiency between the capability and the task specification. The details of this specificity-matching function are given in the next section.

B. Specificity-Matching Function

The function r_{ij} is the specificity-matching function that determines the suitability of an agent to tackle a given task. Based on a symbolic coding scheme for representing the capability of an agent and the characteristics of a task, our framework adopts a string-matching function to determine the best match of a task to the set of available agents. Fig. 2 illustrates the capability coding scheme adopted for an agent capability chain and a task complexity chain. The detailed coding of these chains are explained in Section VI.

Taking reference from Fig. 2, the components of (8), relative intelligence (R_{I}), relative matching length (R_{L}) and relative score (S), are defined as

$$R_{\mathrm{I}} = \frac{\max[I_j]}{\max[C_i]}$$

$$R_{\mathrm{L}} = \frac{l_j}{l_i}$$

$$S = \frac{S_j}{S_i}. \quad (9)$$

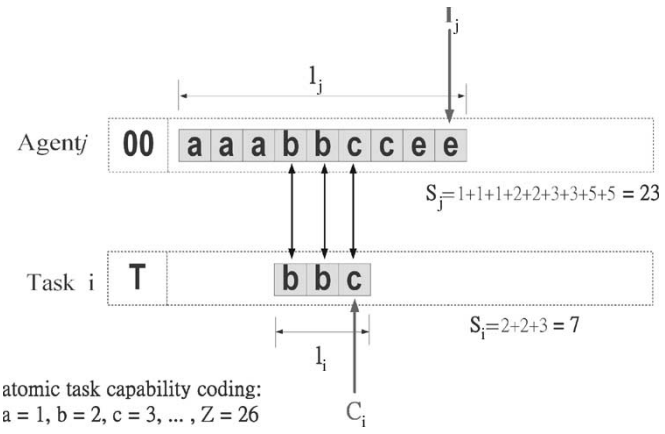


Fig. 2. Parameters for evaluating the specificity-matching function based on the agent capability chain and task complexity chain.

V. Cooperation Model

Cooperation of AIS agents is a result of communications and simple interactions. The design of the AIS-based cooperation model is inspired from its biological counterpart. The core idea of the model follows the cooperation of immune cells, which involves no predefined plans to coordinate team activity. Immune cells work together by mutual understanding through simple signaling. Hence, the AIS agents interact and cooperate with each other through communication.

A. Communications Among AIS Agents

The communication of the AIS-based control framework includes exchanging local information, messaging cooperation signals, and transferring capability between the agents. The communications processes are primarily governed by the internal behavior of the agents. Different behavioral states and their corresponding strategies of the AIS-based control are given in Table II. These behavioral states are analogous to the characteristics and mechanisms of the immune system. The behavioral states allow agents to notify others strategies during operations. Hence, these behavioral states form the basis for achieving a cooperative task through interagent communication. The state-transition diagram given in Fig. 3 underpins how an individual AIS agent behaves and executes different actions in response to the dynamic environment during operations.

AIS agents cooperate by signaling each other through the "request for" and "respond to" help strategies. This signal transmission is known as cooperation by activation and suppression. An agent that is working on a cooperative task and requires help from others is called an initiating agent. An agent that receives a request signal from an initiating agent and responds to the signal is called a responding agent. The initiating agents request for help by sending a "help" signal to all the agents within its CR regardless of whether the agent is capable of offering help. On the other hand, there are no particular rules or algorithms defined to guide and to restrict the agents in responding to the "help" signal. This concept is similar to the human immune system where the immune cells are activated regardless of the type of antigens. The immune cells recognize a wide variety of antigens in the native conformation, without any requirement for antigen processing or display by a specialized system [9].

TABLE II
Behavioral States of AIS Agents

Behavioral State	Notation	Strategy
Achieve	Ach	To tackle the targeted task.
Agitate	Ag	A task has been found and the agent approaches the targeted goal.
Cooperate	Co	To offer help and participate in a cooperative work.
Disperse	D	To keep away from other agents if the number of agents is higher than a threshold value in a wandering zone. A wandering zone is defined as a region where no tasks can be detected.
Explore	E	To explore the surrounding environment and search for tasks randomly.
Idle	I	To wait for other agents for help in tackling a task with cooperation.
Reply	Rp	To receive the 'help' signal from other agents.
Request	Rq	To request for help from other agents.

The main motivation of cooperation among AIS agents is the notification of behavioral states. The identification of the current behavioral state of a responding agent is vital for verifying the feasibility of that agent's capability to offer help. An agent who is in the explore or disperse state is capable of offering help and will be activated by the "help" signal. Contrarily, an agent who is neither in the explore or disperse state will ignore the request and continue its current job. In the meantime, the initiating agent of the cooperative work is being suppressed and change to the idle state until other agents have responded to the request. To allow such cooperation, communication among AIS agents is vital. A formal mathematical description for the communication mechanism of the proposed AIS control is given below.

The jth agent $a_j^{\mathrm{CR}_x}$ that is within the CR of the agent $x(\mathrm{CR}_x)$ is defined as

$$a_j^{\mathrm{CR}_x} \triangleq \left|d_{xj(M)}\right| < \mathrm{CR}_x \tag{10}$$

where $d_{xj(M)}$ is the Manhattan distance between agents x and j, as given by (6).

The set of agents A^{CR_x} that is within the CR_x of agent x is defined as

$$A^{\mathrm{CR}_x} = \left\{a_j \in a | a_j^{\mathrm{CR}_x}\right\} \tag{11}$$

$$\exists A^{\mathrm{CR}_x} \in a, \qquad a_j^{\mathrm{CR}_x} \wedge \neg(a_x). \tag{12}$$

The "help" signal h_x^i sent by agent x for the ith goal is defined as

$$h_x^i = (a_x, g_i) \qquad \forall A^{\mathrm{CR}_x}. \tag{13}$$

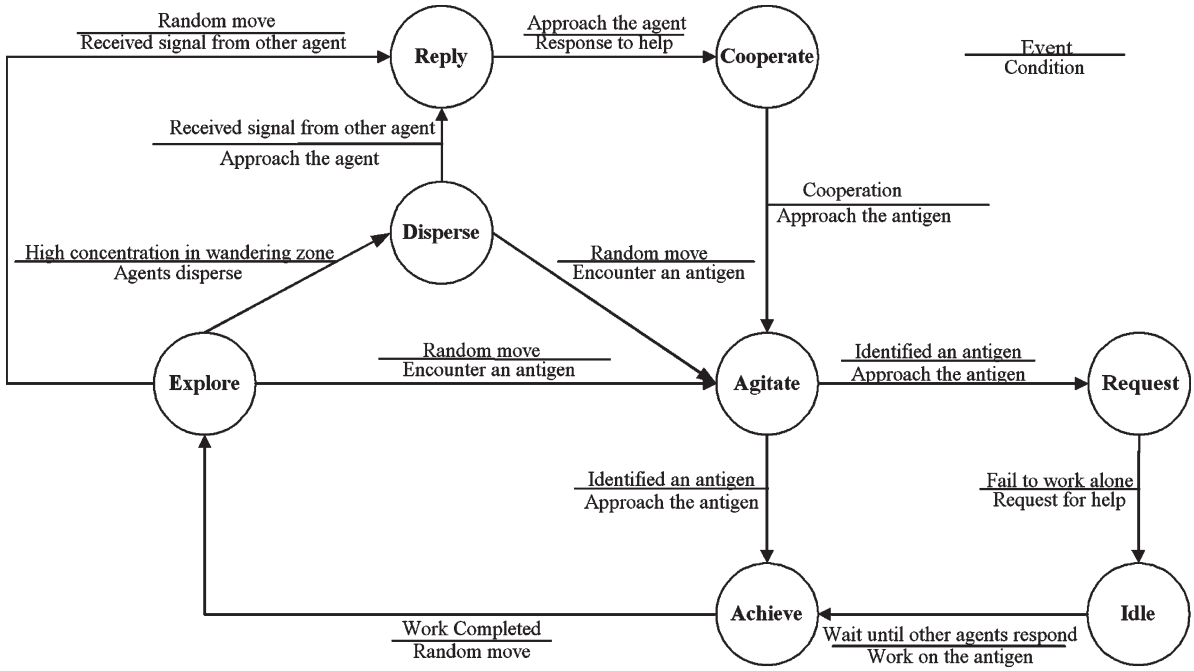


Fig. 3. State-transition model of an AIS agent.

The "response" signal r_j^i sent by agent j with respect to agent x request is defined as

$$r_j^i = \left(a_j, g_i, h_x^i\right) \qquad \forall a_j^{\mathrm{CR}_x}(E \vee D). \tag{14}$$

B. Interaction Between AIS Agents

Interactions between AIS agents can be investigated through the changes of their behavioral states strategically. There are two types of interactions between AIS agents, namely concentration constraint and cooperation. These two interactions are explained in this section.

As mentioned in Table II, the number of agents that are present in a wandering zone is restricted. In practice, this is to maximize the searching efficiency and to avoid overcrowding of agents at any instance. The concept of concentration constraint is inspired by the self-regulation theory of the human immunity. The theory provides a regulatory mechanism for controlling the antibody concentration via stimulation and suppression to the immune response [15]. The immune system produces antibodies when an antigen is presented and returns to equilibrium when antigens have been eliminated.

Assuming there are n number of agents exploring in the environment, and the concentration threshold within a wandering zone is τ, the strategic changes of behavior of the agents under the concentration constraint are as follows

$$na(E) \rightarrow (n-\tau)a(D) \wedge \tau a(E) \qquad \text{if } n > \tau \tag{15}$$

$$na(E) \rightarrow na(E) \qquad \text{if } n \leq \tau. \tag{16}$$

When the number of exploring agents is greater than the concentration threshold within a wandering zone (15), only τ number of agents is in the explore state and searches for a job in their proximities. The reminding $n-\tau$ number of agents will change to the disperse state and move away from the wandering zone. On the other hand, if the number of exploring agents is less than or equal to the concentration threshold, all the agents will stay in the explore state, as given in (16).

Another significance of the introduction of the behavioral states is to illustrate the cooperation between AIS agents. The pervious section has defined cooperation of the AIS agents through communication. With the aid of the notion of behavioral state, strategic behavioral changes in an agent allow different actions to be preformed. For example, when agent x cannot complete the targeted task and it needs to cooperate with other agents, agent x will then send a "help" signal to activate the agents within its proximity. Agent x is therefore in the request state $a_x(\mathrm{Rq})$. The mathematical definition of the "help" signal is given in Section V-C below. In order to be activated by agent x, responding agents must be within the CR of agent x and in the explore or disperse state. The responding agent $a_j^{\mathrm{CR}_x}$ that may offer help and reply to agent x is defined as

$$\forall a_j(\mathrm{Rp}) \in a_j^{\mathrm{CR}_x}, \qquad a_j(E \vee D). \tag{17}$$

The cooperation activity with strategic behavioral control between the AIS agents are defined as

$$[a_x(\mathrm{Rq}) \wedge a_y(\mathrm{Ry})] \rightarrow [a_x(\mathrm{I}) \wedge a_y(\mathrm{Co})] \tag{18}$$

$$[a_x(\mathrm{I}) \wedge a_y(\mathrm{Ag})] \rightarrow [a_x(\mathrm{Ach}) \wedge a_y(\mathrm{Ach})]\,. \tag{19}$$

Equation (18) describes the instance when a "help" signal is transmitted by agent x. Agent x is in the request state when the signal is being sent. Thereafter, agent x changes to the idle state while waiting for help. The responding agent, agent y, is in the reply state when it receives the signal and changes to the cooperate state to offer help. Equation (19) describes the instance when agent y participates in the cooperation activity.

Initially, agent x is in the idle state while waiting for help, whereas agent y is in the agitate state and approaches the cooperative task. After agent y arrives in the task-tackling area, both of the agents complete the task cooperatively in the achieve state.

C. Affinity Threshold

The "help" signal, as mentioned in the previous sections, is a signal transmitted by the initiating agents for the initiation of cooperative work. Here, we define this signal analytically. An initiating agent sends a "help" signal to all its surrounding agents regardless of how many agents will respond to join the cooperation activity. Hence, we introduced an affinity-threshold function to determine whether a responding agent is qualified for the participation of the cooperative work.

Affinity threshold (K_i) is the value that determines whether to activate or suppress a responding agent's activity towards a cooperative task i. Once an exploring or dispersing agent receives a "help" signal, the agent will change to the reply state, as stated in Fig. 3. The agent will then enumerate the binding affinity for the signaled cooperative task i. The affinity threshold is, therefore, a function of the following parameters.

1) β_{ij}—the binding affinity of agent j on task i.
2) η_{ij}—the total number of qualified agents that has detected task i. The qualified agent represents an agent that is capable of tackling task i. This is determined by the computation of the associated binding affinity in which specificity-matching function suggests the fitness of an agent for tackling a particular task. Since all agents in the reply state will respond to the "help" signal, only those agents who are able to tackle task i will change to the cooperate state. Hence, η_{ij} is the total number of agents that are in the cooperate state due to task i.
3) η_{req}—the total number of agents that is required to tackle task i. This parameter specifies the number of agents required to tackle task i

$$K_i = \frac{\sum_j \beta_{ij}}{\eta_{ij} \cdot \eta_{\text{req}}}. \tag{20}$$

In the proposed model, K_i is a dynamic value that controls the activation or suppression of an agent in dealing with task i in accordance with its prevailing affinity index. When β_{ij} of agent j exceeds the value of K_i, the activity is activated. Agent j is said to be activated by the "help" signal. Agent j will then change from the reply state to the cooperate state, approach task i, and tackle the task cooperatively. On the contrary, if the value of β_{ij} of agent j does not exceed the value of K_i, the activity is suppressed. Agent j will change from the reply state back to the explore state and search for other tasks.

VI. Response Manipulation of the AIS-Based Control Framework

The AIS-based control framework presented in Fig. 1 follows the idea of the biological immune responses. Generation of an immune response involves a sequence of actions. They include antigen recognition, activation of lymphocytes, elimination of antigen, and memory [26]. The core of the control framework is to manipulate different responses towards various problems encountered by an AIS agent. By drawing analogy from biological immunity, four main responses have been identified in the response-manipulation algorithm, as specified in Fig. 1. They are nonspecific, acquired, specific, and passive responses.

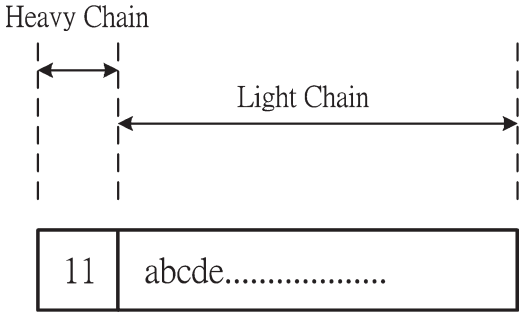


Fig. 4. Capability chain.

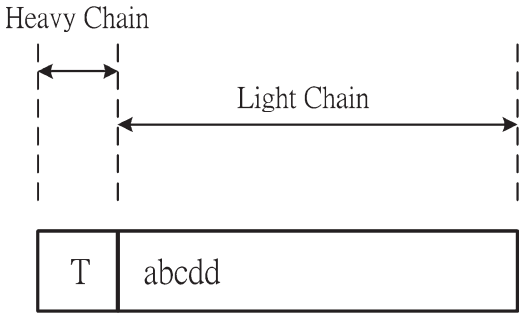


Fig. 5. Task complexity chain.

A. Capability Chain

The basic structure of all antibodies consists of two pairs of heavy and light chains. The heavy chains denote the class of antibody, whereas the light chain is antigentically distinct and only one type of light chain is present in any single antibody molecule [27]. By imitating the structure of an antibody, a pair of heavy and light chains is abstracted to form a capability chain for each of the responses identified in the control framework. The heavy chain specifies the class of responses and is used as an indication for the respective response. The light chain consists of a sequence of atomic abilities. This sequence is distinct for every response. On the other hand, a task, such as a parcel to be moved, is represented by a task complexity chain. It has the same structure as the capability chain, where the light chain defines the task complexity. The complexity specifies the required capabilities of an agent to handle the task. In order to verify the feasibility of an agent to handle a task, the specificity-matching function of (8) is used to match the light chains of both the capability chain and task complexity chain. The structure of a capability chain and a task complexity chain is shown in Figs. 4 and 5, respectively.

Fig. 6 defines how an AIS agent determines the appropriate response autonomously when a task is encountered. Tasks are

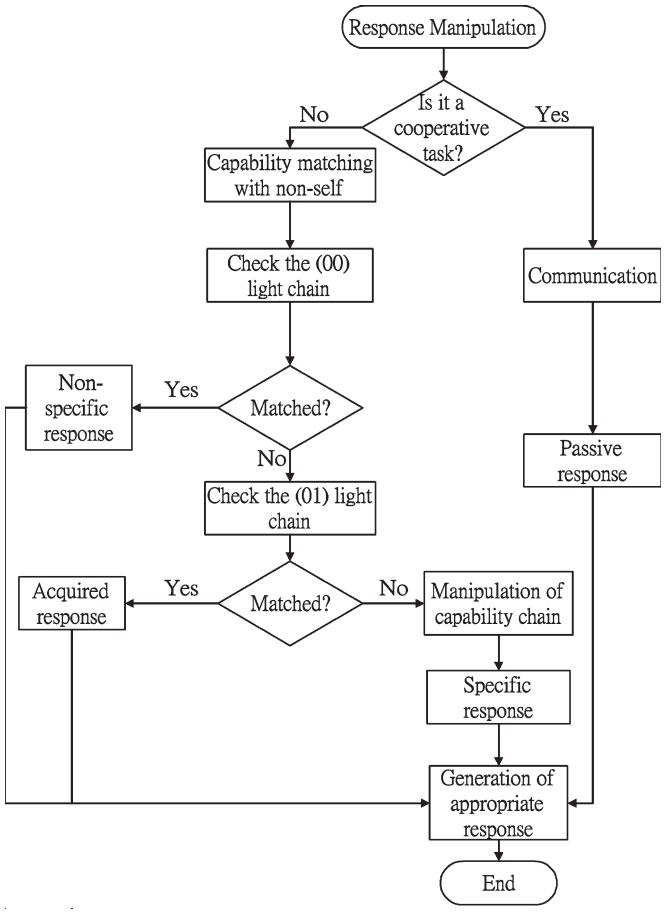


Fig. 6. Overview of the response-manipulation algorithm.

categorized into ordinary and cooperative tasks. An ordinary task can be handled by a single agent, whereas a cooperative task requires more than one agent to handle. In the response-manipulation algorithm, the agent first determines whether it is going to deal with an ordinary task or to respond to a request sent by other agents. If the task requires cooperation, the agent will communicate with the initiating agent and perform a passive response. On the contrary, if the agent encounters an ordinary task, the agent will first check the capability of the nonspecific and acquired chains. If nothing can be matched, the agent will then manipulate its atomic abilities in the nonspecific chain to perform a specific response.

The interrelationship of the four capability chains is shown in Fig. 7. The nonspecific response deals with general and frequently occurring tasks. The sequence of atomic abilities is predefined during system development. Different types of systems will have different encodings for atomic abilities that specify the agents' actions. No capability manipulation is required for nonspecific response. This response is similar to an innate immunity, which is the first general defense that provides resistance to antigens. For a distinct and specific task, the agents will generate a new set of capabilities through capability manipulation. This manipulation includes recombination of atomic abilities from the nonspecific response. The new set of capabilities will then go to the acquired response chain. This mechanism is equivalent to secondary immunity, where a faster and stronger response will result in the next occurrence of the same antigen.

B. Symbolic Coding Scheme for the Chains

Capabilities of AIS agents are classified into two main categories. They are simple capability and compound capability. Simple capability enables basic and straightforward actions to be performed whereas compound capability deals with more complex and difficult tasks. Simple capability is represented by fundamental atomic ability such as "a," "b," "c," and "d." Compound capabilities are set of capabilities generated from the simple capability. For example, a set of compound capabilities generated from "a," "b," and "c" may include: {"aa," "aaa," "abb," "abc"} and {"bb," "bbb," "baa," "bca"}, etc.

Every task assigned in a workplace is specified with a complexity or a chain of complexities. This complexity is a chain of code, similar to the capability chain that indicates what capabilities are required by an agent to tackle the task.

Our control framework is designed for a range of multiagent applications. The symbolic coding scheme is generic to different kind of applications. Since simple string representation is used, the codes assigned for the agents' capability and the task complexity is interchangeable, which depends on the requirements of the application. In order to better understand the internal representations of capability chains and task complexity chains, the symbolic coding scheme is illustrated with examples of a material-handling system.

1) Material-handling task—{Task complexity chain}
   a) Grouping of goods—{g}
      The complexity "g" notifies agents that the task is a grouping task. Agents are programmed to group all goods with a complexity "g" together in a given workplace.
   b) Counting of goods—{c}
      The complexity "c" notifies agents that the task is a counting task. Agents are programmed to count all the goods with a complexity "c" at a given workstation.
   c) Goods reallocation—{aabf}
      The first part of the complexity "aa" notifies agents that the task is a goods reallocation task. The second part "b" indicates that the goods need to be reallocated, and lastly, the third part "f" indicates the location where the goods are going to be allocated. This reallocation task is expected to be performed within the same workstation. If goods are required to be allocated to a different workstation, the task is classified to be a goods-delivery task.
   d) Goods delivery—{ddaabzE}
      The first part of the complexity "dd" notifies agents that the task is a goods-delivery task. The second part "aa" specifies that the goods are going to be reallocated. Combing the first two parts, we have "ddaa," which means the goods are going to be reallocated in a different workstation. Hence, agents need to deliver those goods to another place. The third part "b" specifies what goods are needed to be delivered and the forth part "z" specifies where the goods are going

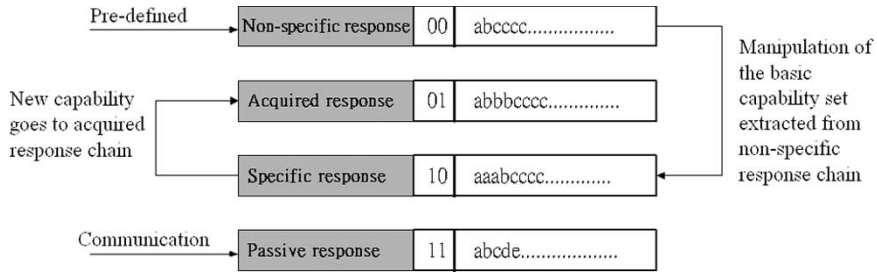


Fig. 7. Interrelationship between the four capability chains.

to be delivered. Lastly, the capital letter at the end of the complexity chain represents that the task requires more than one agent to handle. In this case, a capital letter "E" states that a total number of five agents are required for this cooperative task.

e) Searching of goods—{ssqk}

The first part of the complexity "ss" notifies agents that the task is a goods-searching task. The second part "q" specifies which kind of goods is going to be searched. The last part "k" specifies the workstation area for this searching task to be performed. On the other hand, if there is no specified location for searching the goods, for example {ssq}, agents then search for the goods anywhere within the workplace.

From the above material-handling examples, all the tasks except task d are referred as ordinary tasks. This is because they only require one agent to complete the tasks. The capital letter at the end the chain indicates that task d is a cooperative task. Hence, to specify a task as a cooperative task, a capital letter is appended at the end of a complexity chain indicating how many agents are needed for that cooperative task. For example, to assign two agents for the grouping task (task a), the complexity chain is changed to {gB}. The capital letter represents that the task is a cooperative task, and for illustration purposes, the alphabetical order of the letter is used to represent the number of agents required to handle the task.

2) Capability chain

a) Agent x—{cg}

With atomic abilities "c" and "g," agent x can perform ordinary activities in the material-handling workplace. Since the capability chain of agent x contains only simple capabilities, it is an agent with low intelligence level.

b) Agent y—{cgddaabfz}

In this example, agent y has a high intelligence level. Agent y contains more capabilities than agent x. The first two atomic abilities of agent y are the same as agent x. Hence, agent y can handle both tasks a and b. The rest of the capability chain represents two sets of capabilities. First, {ddaa} represents that agent y is able to perform a goods-delivery task. Second, {aa} represents that agent y is able to perform a goods-allocation task. The atomic ability "b" indicates the type of goods. The last two atomic abilities "fz" indicate the locations for goods allocation or delivery. For example, the set of capabilities {aabf} allows agent y to perform task c. On the other hand, if agent y is asked to perform task d, which requires a set of capability {ddaabz}, capability manipulation is executed to generate a new set of capabilities for agent y to tackle task d. This capability manipulation recombines the original set of capabilities {ddaabfz} of agent y to a new set of capabilities {ddaabzf}. Hence, the new set of capabilities that consists of {ddaabz} indicates that agent y is now able to tackle task d. The procedures for capability manipulation are given in Section VI-C.

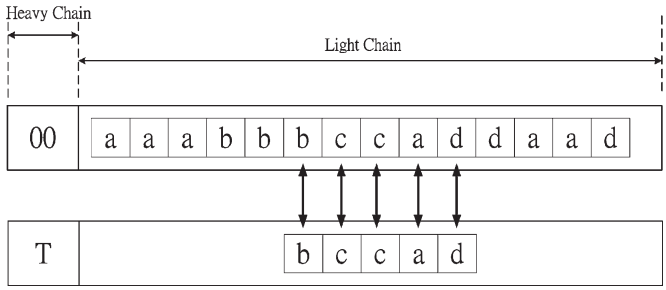


Fig. 8. Absolute string matching (I).

C. Methodology of Specificity Matching

String representations and different kinds of string-matching algorithms have been used extensively in the research of AIS [28]–[31]. The specificity matching utilizes the string-matching algorithm in [32]. This specificity function is used to determine the feasibility of an agent to handle a given task. In immunology, the activation of immune cells, namely lymphocytes, is triggered by the recognition of antigens [26]. The lymphocytes recognize antigens structurally by their antigen receptors. Following this pattern-recognition concept, a string-matching algorithm is developed to match tasks with agents according to the corresponding complexity and capability chains. Hence, the light chains of both the agent and the task are compared to determine if the agent is competent to handle the targeted task.

In specificity matching, nonspecific (00) and acquired (01) capability chains are first being investigated since they provide faster responses by an absolute string-matching method, as illustrated in Figs. 8 and 9. The absolute string-matching algorithm determines if the whole string of a task-complexity light chain can be found from an agent's capability light chain

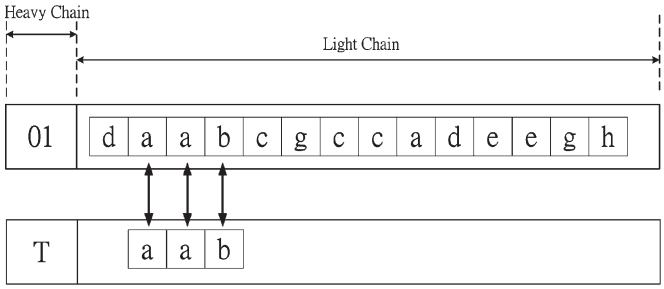


Fig. 9. Absolute string matching (II).

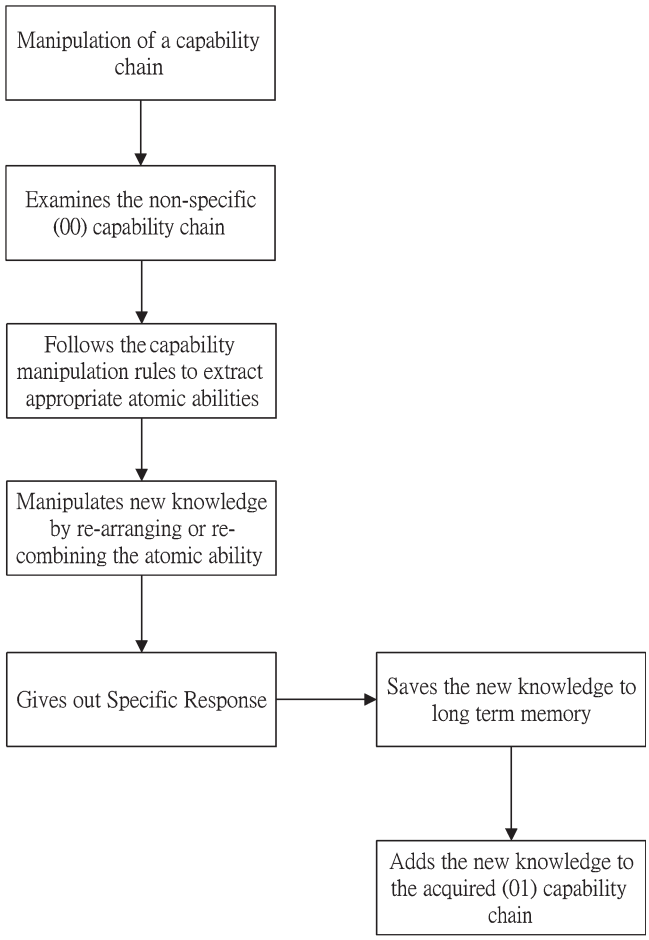


Fig. 10. Sequence of the capability-manipulation process.

and gives a definitive result of either the two chains are matched or mismatched.

When a mismatch results from both nonspecific and acquired responses, a specific response will then mediate to manipulate new knowledge that is capable of tackling the task. The capability manipulation recombines or rearranges the elementary atomic abilities of the nonspecific capability chain to generate a new set of capabilities that is specific to the new task. This process resembles immature lymphocytes having receptor-editing function, where receptors with self-specificity can make further rearrangements of their light chains [33]. Fig. 10 shows the stages of the capability-manipulation process.

A set of capability-manipulation procedures are used to determine how the atomic abilities in the nonspecific (00) capability chain are manipulated to form new sets of capabilities. Not all generation of new capabilities will be successful. The atomic abilities of the nonspecific capability chain should be qualified for a recombination. All the capability-manipulation procedures must be satisfied. The procedures for capability manipulation are given as follows.

Step 1) Disjoin the antigen complexity light chain into separate required capabilities.

Step 2) Compare the disjointed required capabilities of the task complexity with the capability chain. Check if all the disjointed capabilities are contained inside the nonspecific (00) capability chain.

Step 3) Generate all possible combinations of the required capabilities. These combinations are in the simplest form with minimum duplication of atomic ability.

Step 4) Compare all the disjointed required capabilities generated by Step 3) with the capability chain. Extract all the matched capability sets.

Step 5) Compute the matched sets by Dijkstra's algorithm to find out the shortest path to generate the new capabilities set. This minimizes the computational steps in rearranging or recombining a new set of knowledge.

The new capability set developed by capability manipulation is appended to the acquired (01) capability chain after the task has been completed. The new string is concatenated to the acquired (01) capability chain by an append-string algorithm. The append-string algorithm is developed to attach the new capability set to the end of the exiting chain without any duplication of atomic abilities. The algorithm is given as follows.

1) If the length of the new capability set newCap is equal to n, get n characters (getChar) from the end of the existing capability chain.
2) Match the strings getChar and newCap by absolute string matching.
3) If nothing is matched, delete the first character of getChar. This is done by shifting $S_{(t-1)}$ the getChar pattern by getChar $= n - (t-1)$, where $t = 1$ is the first matching.
4) Match getChar with newCap by prefix string matching until $t = n$. This prefix string-matching function will return true (matched) only if the whole getChar pattern matches the prefix of the newCap.
5) When a matched is found, delete the first $(n - S)$ characters of newCap and append the newCap to the end of the exiting capability chain.
6) If no match is found, append the whole newCap to the end of the existing capability chain.

The procedures of adding new capabilities by the append-string algorithm are illustrated diagrammatically by Figs. 11 and 12. In Fig. 11, nothing has been matched till $t = n = 4$, therefore, the whole set of new capabilities is appended to the end of the existing capability chain. On the other hand, in Fig. 12, nothing has been matched when $t = 1$. After the

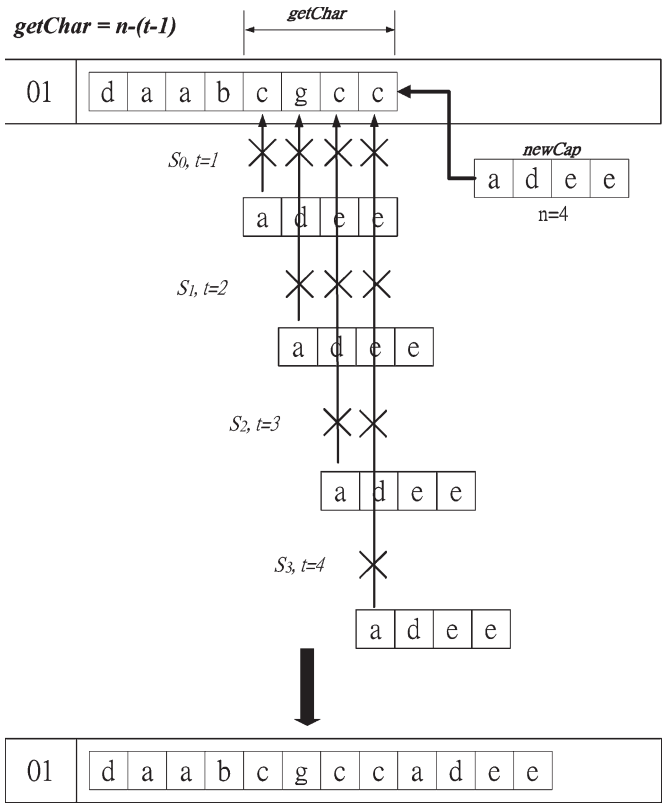


Fig. 11. Addition of new capability (case 1).

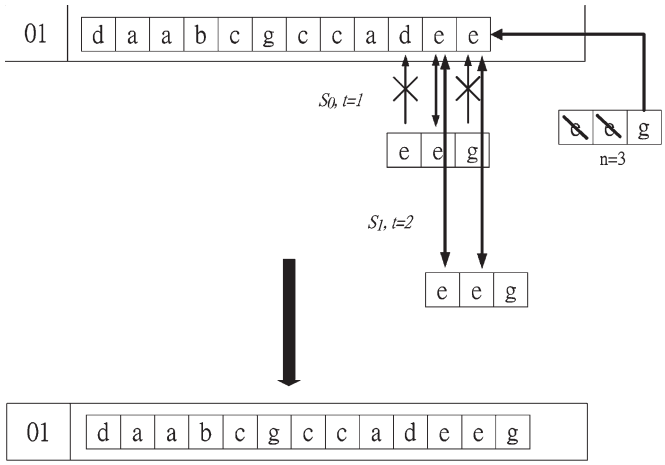


Fig. 12. Addition of new capability (case 2).

first shift (S_1), getChar becomes an "ee" pattern. A match is found when the getChar pattern is matched with the prefix of the new capability set. Hence, the first two characters of getChar, which are "ee," are eliminated from the new capability set. The remaining pattern of the new capability, which is the "g" pattern, will then be appended to the existing capability chain.

D. An Illustrative Example of Response Manipulation

The response manipulation introduced in the previous sections is illustrated by an example. We focus on how an individual AIS agent determines the activation of different responses. For simplicity, cooperation among agents, indicated by the passive (11) capability chain, is not considered.

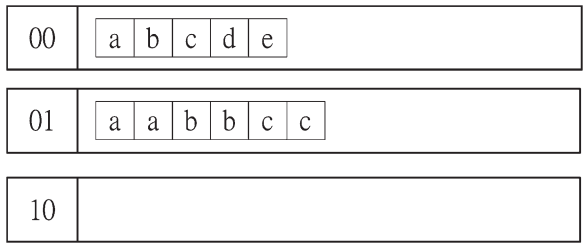


Fig. 13. Capability chains of agent j.

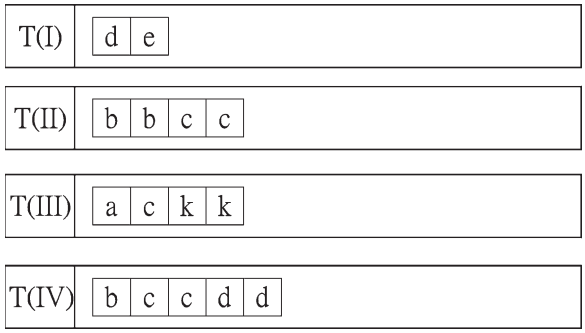


Fig. 14. Task complexity chains of tasks (I)–(IV).

An example is presented to illustrate the operation of the response manipulation under the proposed scheme. To consider the generic operation of the response-manipulation scheme, the actual domain of application of the capabilities is not considered in this example. Assume the capability chains of agent j and the task complexity chains of tasks I–IV are given by Figs. 13 and 14, respectively.

To tackle a task, the AIS agent first investigates the nonspecific (00) and acquired (01) capability chains with absolute string matching. For task I, a match is found in the nonspecific (00) chain. A direct and simple nonspecific response "de" is then carried out. task II requires a set of capabilities "bbcc" to tackle the problem. As no match can be found on the nonspecific (00) capability chain, the acquired (01) chain is then examined. When a match is found, the AIS agents then execute a direct and rapid "bbcc" acquired response. The pseudocode for response manipulation is given in Fig. 15.

On the other hand, task III requires a set of capabilities "ackk." Since this capability set cannot be found in both nonspecific and acquired chains, the AIS agent needs to manipulate its capability in order to complete the task. The capability-manipulation procedures presented in Section VI-C are then followed. The capabilities "ackk" are disjointed and becomes "a," "c," "k," and "k." According to Step 2), all the disjointed required capabilities are referenced from the nonspecific (00) capability chain to allow capability manipulation. Atomic capabilities "a" and "c" are found; however, there are no capability "k"s available in the nonspecific (00) chain. Hence, this set of capabilities "ackk" is not available to perform the required response through specific capability manipulation. The AIS agent has therefore failed to handle task III.

```
nonChain = capabilities of the non-specific (00) capability chain
acqChain = capabilities of the acquired (01) capability chain
taskChain = the capacities of antigen complexity chain

match = absoluteStringMatch(taskChain,nonChain)
if match = TRUE
   nonSpecificResponse()
else
   match = absoluteStringMatch(taskChain,nonChain)
   if match = TRUE
      acquiredResponse()
   else
      capabilityManipulation()
```

Fig. 15. Pseudocode of the response-manipulation algorithm.

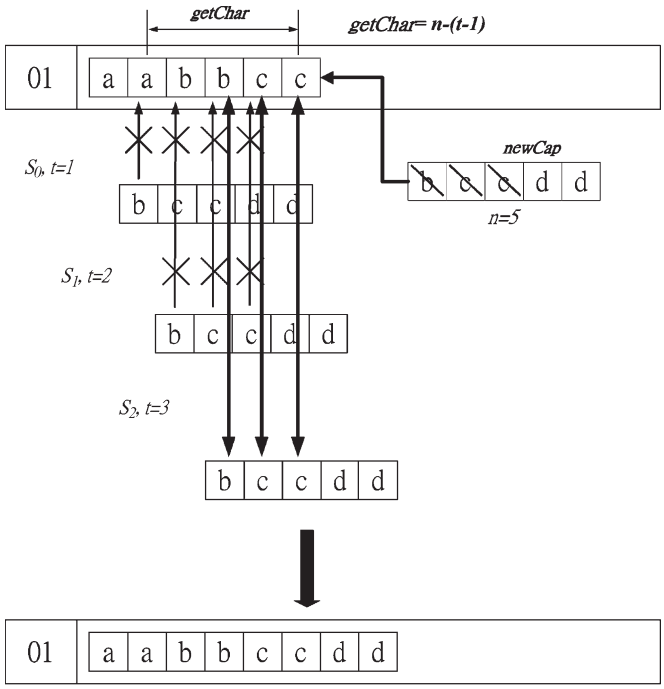


Fig. 16. Acquired (01) capability chain of agent j after capability manipulation.

Similar to task III, the required capabilities of task IV "bccdd" need to be manipulated. The disjointed required capabilities are "b," "c," and "d," which is a legal set of capabilities for manipulation. To rearrange a new set of capabilities that is specific to task IV, the "bcd" pattern from the nonspecific (00) capability chain is extracted for manipulation. Newly rearranged patterns such as "bc," "cd," "bd," "bbc," and "cdd" are generated. By applying Dijkstra's algorithm, patterns "bc" and "cdd" are recombined to form the new capability set "bccdd." In this case, the new capability set provides specific response to solve task IV.

The new capability "bccdd" is then appended to the acquired (01) capability chain. Following the append-string algorithm given in Section VI-C, "bccdd" is the newCap with n equal to 5. When $t = 1$ and $S = 0$, getChar is "abbcc" and a matching string is found when $t = 3$ and $S = 2$. As there is a match, the first $(n - S)$ characters, which are the first three characters "bcc," are deleted from newCap. The finalized newCap "dd" is then appended to the acquired (01) capability chain without any duplication (Fig. 16).

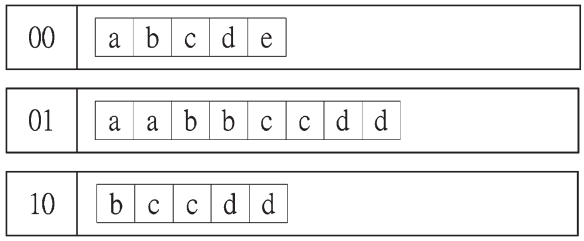


Fig. 17. Capability chains of agent j after completing tasks (I)–(IV).

After completing tasks I–IV, the capability chains of agent j are shown in Fig. 17. The new capabilities acquired from handling task IV are added to the acquired (01) capability chain. This leads to a stronger and faster response in future reoccurrence of task IV. Having different capability chains to perform response manipulation, AIS agents are able to perform rapid responses by issuing corresponding actions to tackle a wide range of tasks. An intelligent AIS agent that is adaptive in tackling new problems can be achieved through the immunologic response manipulation.

VII. Operational Scheme

A. Control Logic

Drawing parallels to the human immune system, the self and nonself of our control framework correspond to the AIS agents and tasks. AIS agents start the operation cycle by undergoing pseudorandom motions. This "free" exploration mode allows the AIS agents to identify and approach nonself. If the number of exploring AIS agents exceeds a threshold that defines the overcrowdedness within a wandering zone, the agents will disperse to maintain efficient exploration. Once a nonself has been sensed by the AIS agents, they utilize binding affinity (5) to recognize and approach these tasks. The binding affinity evaluates the physical factors and specificity of the agent to a given task. The higher is the binding affinity, the better is the suitability of the agent to tackle the task. The control logic of the control framework is given in Fig. 18.

B. Behaviors of AIS Agents

AIS agents alter their behavior by monitoring the dynamic environment. Different behavioral states and corresponding strategies are given in Table II. To tackle a task, AIS agents

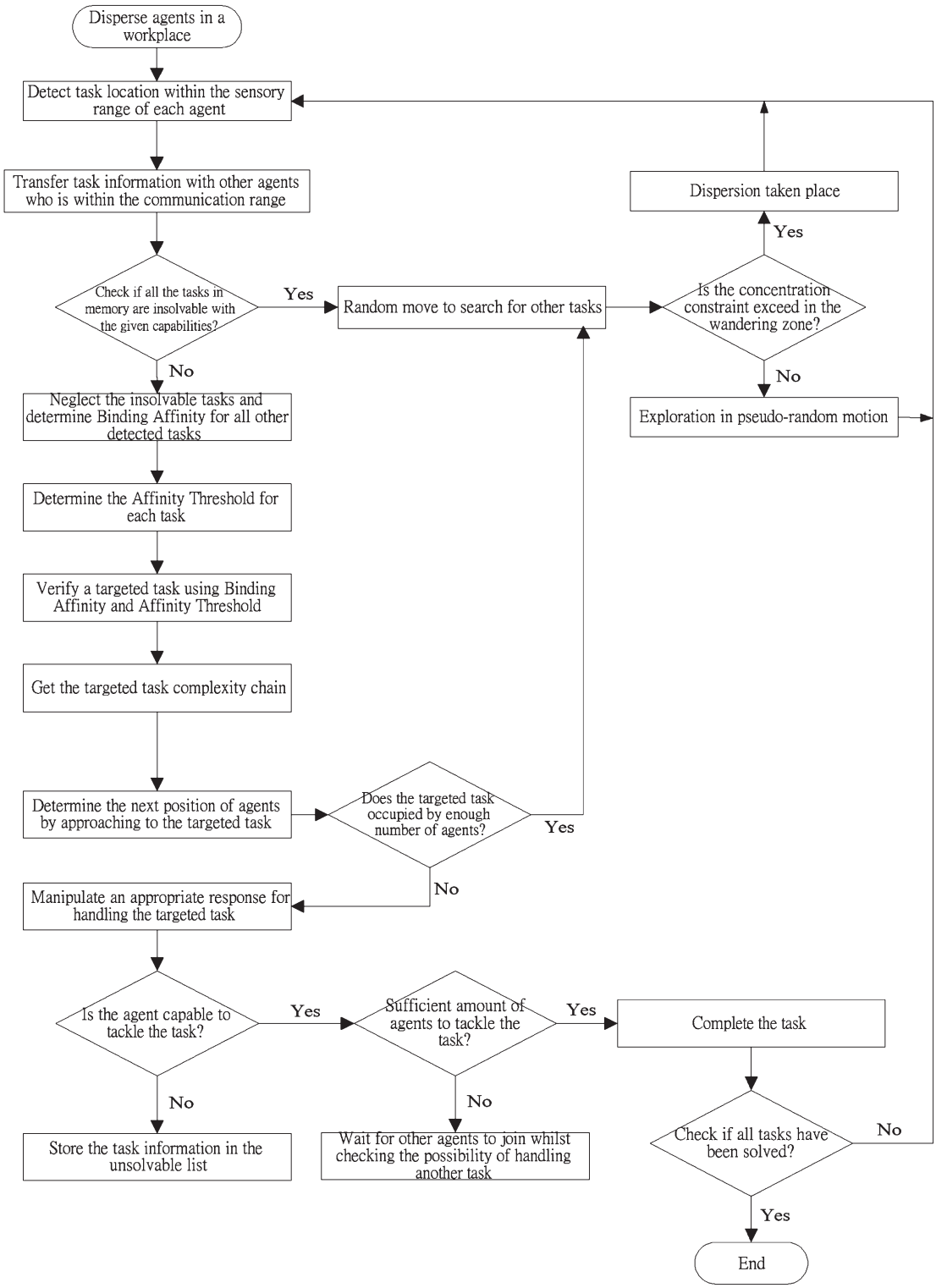


Fig. 18. Control logic of the AIS-based control framework.

exhibit different behaviors. This is represented by the state-transition model shown in Fig. 3.

The state transitions in the model are governed by a set of event–condition–action rules. These rules allow AIS agents to perform corresponding responses under different situations. According to these rules, AIS agents obtain information through interagent communication. This information includes task locations sensed by other agents, a "help" signal sent by an initiating agent, and behavior states of other agents, etc. All of these are useful information that allow agents to coordinate plans of action for different kinds of tasks in a cognitive manner. For example, when agent a is tackling a cooperative task, agent a will send a "help" signal to all its surrounding agents. Only agents that are in the explore and disperse states will reply

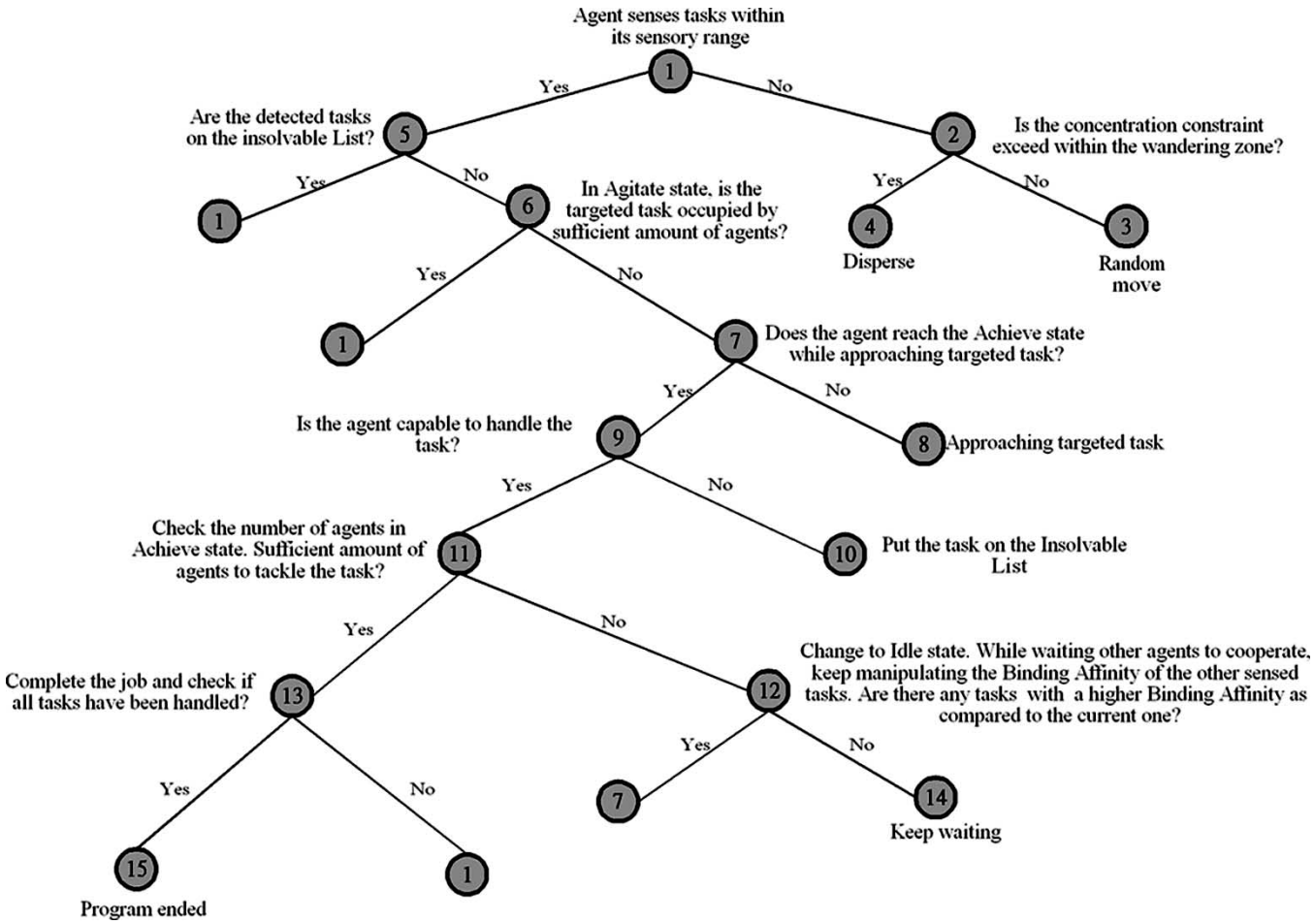


Fig. 19. Decision tree defining the actions of the AIS agent in response to different situations.

to agent a. On the other hand, when the responding agents have approached the cooperative task, they will change to the achieve state and are ready to tackle the task. Hence, agent a can evaluate if there are enough number of agents presented for the cooperative work by taking information of their behavioral states. When the number of agents that are in the achieve state is enough for the cooperative task, agent a will stop sending the "help" signal. The group of cooperating agents will then complete the task together.

Fig. 19 shows the decision tree that defines the actions of the AIS agents. The decision tree shows how the AIS agents will adapt to the changing environment and behave autonomously in tackling tasks.

C. Simulation Study

To demonstrate the effectiveness of the proposed control framework, simulation studies are conducted. In the simulation, groups of AIS agents are deployed in a warehouse. Tasks assigned in the warehouse are predefined with different complexity levels. For example, counting and building up of goods in the warehouse are of simple complexities and regarded as simple tasks. Other tasks in typical warehousing operations include goods storage, retrieval, compartment assignment, etc., are of higher complexity levels and are regarded as more difficult tasks. The performance of the control framework is being studied in the following two aspects.

1) Binding Affinity—the AIS agents in the first simulation are activated only by the nearest task. They only concern the distance between the task and themselves. In the second study, binding affinity is evaluated according to (5) together with the affinity threshold (20), by which AIS agents can either be stimulated or suppressed. In this case, their activities depend on their specificity, distance, and familiarity with the task encountered.
2) Intelligence of AIS agents—with the same set of tasks being investigated in all the simulation cases, agents of different capabilities are employed to handle the tasks. This is to demonstrate the importance of response manipulation, which involves the consideration of specificity between an agent and a task.

Three cases with different input parameters have been given in Table III. The affinity measures of cases II and III follow the model derived from the control framework. To highlight the efficiency of the response manipulation, agents of different intelligence levels are being tested. The intelligence levels are determined by the simple and compound capabilities introduced in Section IV. In this simulation, agents that are encoded

TABLE III
THREE CASES WITH ASSOCIATED INPUT PARAMETERS FOR THE SIMULATION STUDIES

Case	Affinity measures	Agents' intelligence level	Agents with simple capability	Agents with compound capability
I	Distance	Varied	75%	25%
II	Distance, specificity and affinity threshold	Varied	75%	25%
III	Distance, specificity and affinity threshold	Same	0%	100%

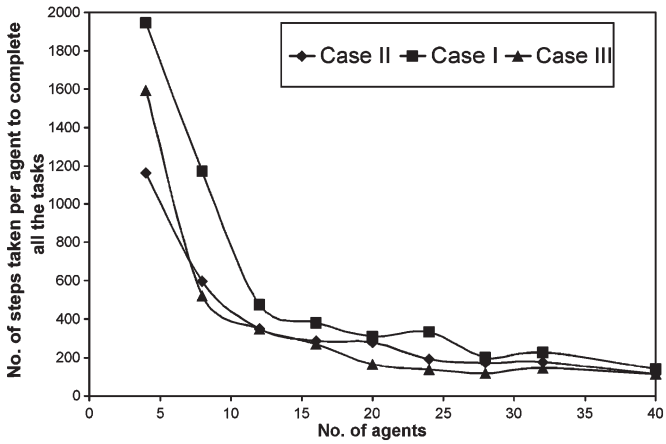


Fig. 20. Performances of the AIS-based control framework.

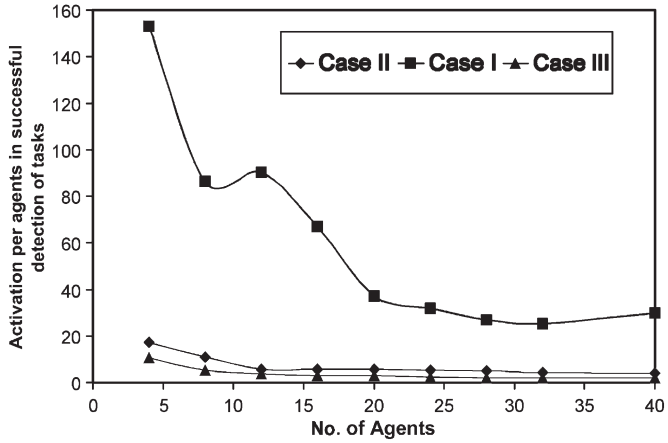


Fig. 21. Average activation level per AIS agent.

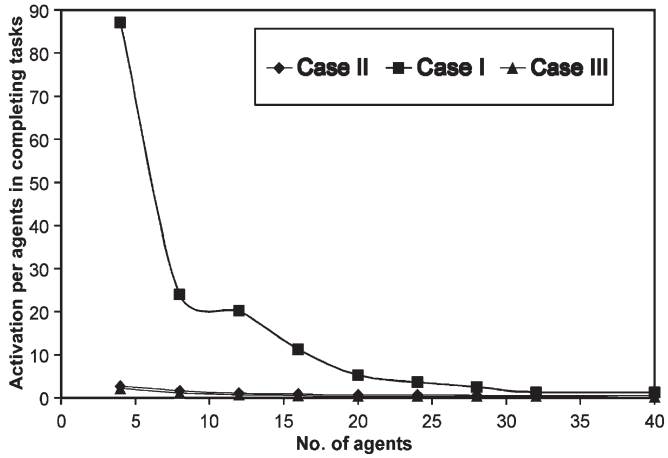


Fig. 22. Time for task completion per AIS agent.

with compound capabilities are capable of tackling all the tasks whereas agents that are encoded with simple capabilities may not be qualified to do so. The agents in cases I and II have different intelligence levels. In these two cases, only a few agents have full capabilities to handle all the tasks. Some of the agents need to perform specific responses by capability manipulation in order to tackle tasks of higher complexities. Agents who failed to perform specific response imply that their nonspecific capabilities are not qualified to generate new capabilities that are suitable to tackle the difficult tasks. On the contrary, all agents in case III have the same set of compound capabilities. They are able to tackle all the tasks assigned. Hence, the overall intelligence level of the agents of case III is higher than those in cases I and II.

Figs. 20–22 present some of the results obtained from the simulation studies. The overall efficiency of the control framework is shown in Fig. 20. All three cases have the same trend in performance. As the number of agents increases beyond 10, the performance of the three cases converges. However, cases II and III are more effective than case I when a smaller number of agents are deployed in the system. The main reason is that in cases II and III, the proposed binding affinity measure is used whereby agents target a task with the best specificity. Moreover, the affinity threshold (20) is computed by taking into account the binding affinities of all the agents that are qualified to tackle a particular task. Looking at this from another viewpoint, the task would attract the most appropriate agent(s), which is unlike case I, in which all the agents in the vicinity are being activated once they have detected a task.

The average activation level of agents that are activated by task detection for each case is shown in Fig. 21. From the results, case I has the highest number of activation per agents. Since the agents in case I are being stimulated regardless of the affinity level, they are solely based on the measure of distance in identifying targeted tasks. A suboptimal level of stimulation that corresponds to unnecessary communication overheads for task detection, therefore, results in case I. A much better stimulation approach is applied in both cases II and III, which is based on the proposed affinity measures (5) and (20). In the simulation study, (5) computes the affinity between a specific agent and its corresponding task, whereas (20) provides the threshold value that is obtained by taking account of all the relevant agents' affinity measures.

Fig. 22 shows the activation per AIS agent in completing tasks. With the same set of tasks being tested, the activation level per agent in case I is much higher than cases II and III before the level of activation to achieve tasks converges (i.e., when the number of agents is more than 20). This is because of the unconditional stimulation received by the agents in case I. All the agents are being activated once a task is detected,

TABLE IV
INPUT PARAMETERS FOR CASES A AND B

Case	A	B
Affinity Measures	(1) Distance (2) Specificity Matching (3) Affinity Threshold	(1) Distance
Agents' Intelligence Level	Same	Same
Agents' initial positions	Centre of the x -axis	Centre of the x -axis
Task Complexity	Same	Same
No. of agents required for:		
Task 1 - (2,6)	3	3
Task 2 - (5,6)	2	2
Task 3 - (8,6)	4	4

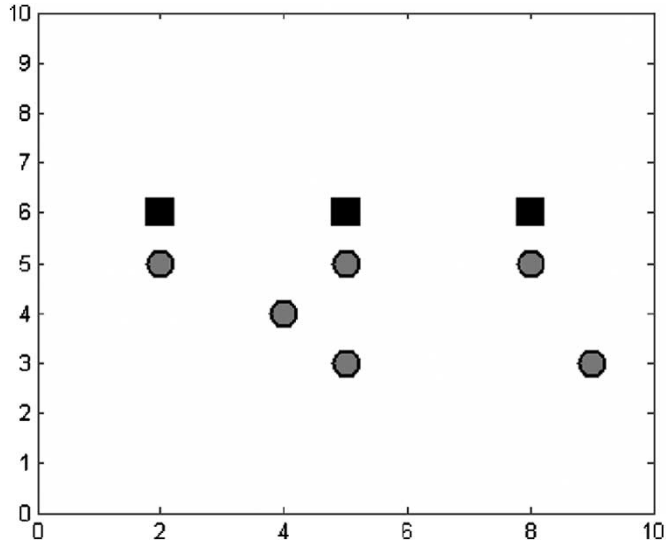


Fig. 23. Case A (i).

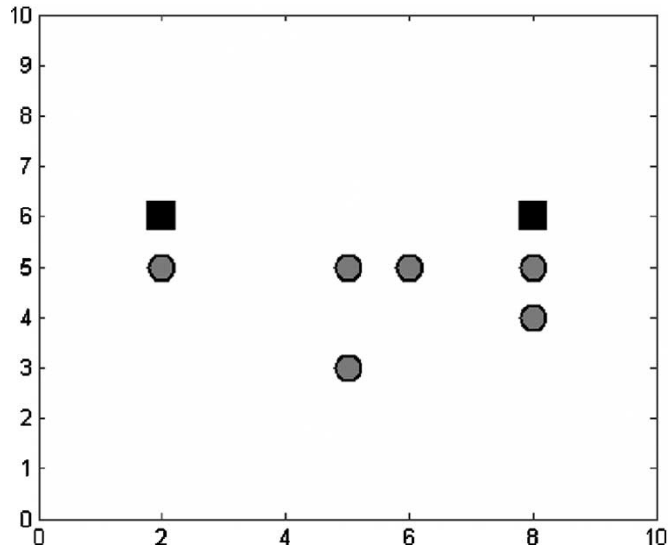


Fig. 24. Case A (ii).

which leads to inefficient job allocation. On the other hand, a stable activation level for task completion (Fig. 22) as well as for task detection (Fig. 21) is obtained for both cases II and III. According to the results, 1.5 activations per agent were obtained in Fig. 22 and five activations per agent were obtained in Fig. 21. This suggests an improved utilization and management of the proposed AIS-based control, where agents are stimulated according to their specificity and other agents' affinity indexes. Unlike the nonspecific organization in case I, where all qualified agents tend to crowd towards a specific task, agents in the proposed AIS-based control framework exhibit self-organized behavior with improved efficiency.

All the agents in case III have compound capabilities. They are able to solve all the tasks assigned to them in the workplace. While in case II, only 25% of the agents have full capabilities to tackle all the tasks. However, the activation levels for task detection and task completion per agent for cases II and III are almost the same. This observation suggested that under the AIS-based control, a system having only a few agents with full capabilities performs comparably well with systems with all agents having full capabilities. The effectiveness of the response-manipulation scheme is therefore demonstrated.

To further investigate the operation of the proposed framework, actual simulation outputs are captured from two new sets of simulations that require cooperation. This study aims to illustrate the effectiveness of the affinity measures as the affinity functions can considerably enhance cooperation between AIS agents. For illustration, a small number of agents and tasks are used in the simulation studies. In these simulations, tasks are represented by square objects and agents are represented by circles, and the input parameters are given in Table IV. Since all the tasks allocated in the workplace are cooperative tasks and have the same complexity level, specificity matching has no significance in this study.

Figs. 23–26 illustrate the distribution of agents at two instances (i) and (ii). Instance (i) was captured in the beginning of the simulation when all the tasks have not been tackled. Instance (ii) was captured when the middle tasks have been tackled. For case A, the agents were evenly distributed in the workplace, whereas in case B, the agents were clustered around one of the task initially. At instance (i), where all the tasks were present in the workplace, all the agents of case B approached task 2 at the same time. This is because the agents of case B only took account of the measure of distance between agents and

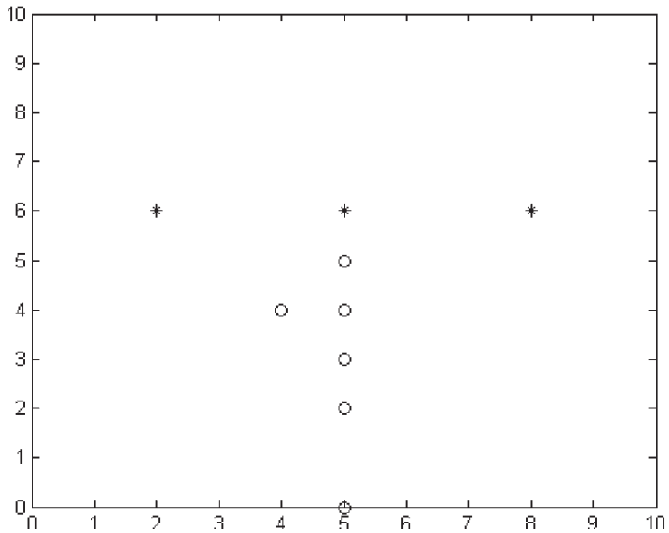


Fig. 25. Case B (i).

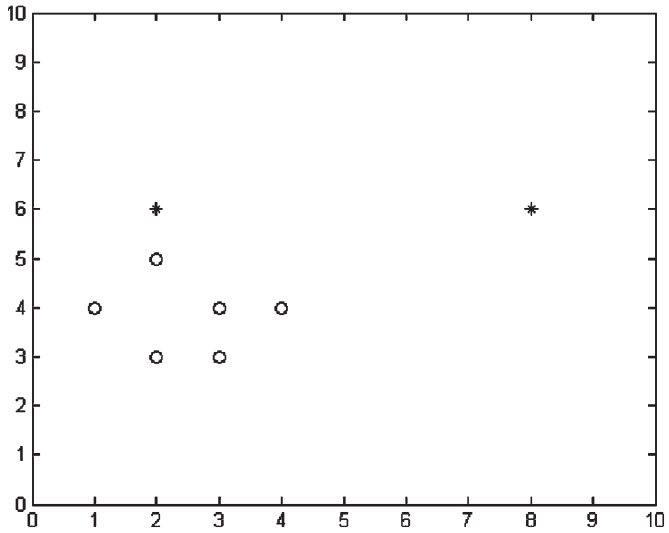


Fig. 26. Case B (ii).

tasks as the affinity measure. On the contrary, the agents of case A considered also the affinity threshold. This affinity threshold allows effective means of cooperation. As only two agents were required to tackle task 2, agents with a lower affinity with task 2 were therefore suppressed from taking part in the cooperation. They will turn to other tasks as there was enough number of agents approached task 2. At instance (ii), all the agents of case B approached task 1 after task 2 was completed, while the agents of case A approached both tasks 1 and 3 in a more disperse manner. Specifically, a larger number of agents in case A at instance (ii) approached task 3. This is because the agents have considered that task 3 required four agents to serve. The affinity between some of the agents and task 3 was therefore higher than task 1. This further demonstrated the effectiveness of the affinity threshold in cooperation.

VIII. Conclusion

This paper presents the development of an AIS-based control framework for the effective control of a group of distributed agents with diverse capabilities. The control framework encapsulates how an AIS agent operates autonomously in a dynamic environment. A mathematical model is presented to formally describe the AIS-based control framework. Through the strategic behavioral study of AIS agents, cooperation is achieved by mutual understanding and communication between them. Furthermore, the response manipulation proposed in the framework establishes the key concept of how an AIS agent can determine its capability in different situations when tackling problems. The work aims to develop a truly decentralized and self-organized multiagent system where agents are adaptive, autonomous, and intelligent in decision making. Simulation studies are presented in the paper to verify the feasibility and effectiveness of the control framework.

References

[1] A. S. Perelson and G. Weisbuch, "Immunology for physicists," *Rev. Mod. Phys.*, vol. 69, no. 4, pp. 1219–1267, Oct. 1997.

[2] T. Fukuda, K. Mori, and M. Tsukiyama, "Parallel search for multi-modal function optimization with diversity and learning of immune algorithm," in *Artificial Immune Systems and Their Applications*, D. Dasgupta, Ed. Berlin: Springer-Verlag, 1998, pp. 210–219.

[3] J. Kim and P. Bentley, "Towards an artificial immune system for network intrusion detection: An investigation of dynamic clonal selection," in *Proc. Congr. Evolutionary Computation (CEC)*, Honolulu, HI, 2002, pp. 1015–1020.

[4] S. Sathyanath and F. Sahin, "AISIMAM—An artificial immune system based intelligent multi agent model and its application to a mine detection problem," in *Proc. Int. Conf. Artificial Immune Systems (ICARIS)*, Canterbury, U.K., pp. 22–31. Session I.

[5] D.-W. Lee, J.-H. Jun, and K.-B. Sim, "Realization of cooperative strategies and swarm behavior in distributed autonomous robotic systems using artificial immune system," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, Tokyo, Japan, 1999, vol. 6, pp. 614–619.

[6] H. Meshref and H. VanLandingham, "Artificial immune systems: Application to autonomous," in *Proc. IEEE Int. Conf. Agents, Systems, Man, and Cybernetics*, Nashville, TN, 2000, vol. 1, pp. 61–66.

[7] L. Sokolova, "Index design by immunocomputing," in *Proc. Int. Conf. Artificial Immune Systems (ICARIS)*, Edinburgh, U.K., 2003, pp. 120–127.

[8] P. Tieri, S. Valensin, C. Franceschi, C. Morandi, and G. C. Castellani, "Memory and selectivity in evolving scale-free immune networks," in *Proc. Int. Conf. Artificial Immune Systems (ICARIS)*, Edinburgh, U.K., 2003, pp. 93–101.

[9] D. W. Bradley and A. M. Tyrrell, "Immunotronics—Novel finite state machine architecture with built-in self-test using self-nonself differentiation," *IEEE Tran. Evol. Comput.*, vol. 6, no. 3, pp. 227–238, Jun. 2002.

[10] A. Ko, H. Y. K. Lau, and T. L. Lau, "An immuno control framework for decentralized mechatronic control," in *Proc. Int. Conf. Artificial Immune Systems (ICARIS)*, Catania, Italy, 2004, pp. 91–105.

[11] R. O. Canham and A. M. Tyrrell, "A multilayered immune system for hardware fault tolerance within an embryonic array," in *Proc. Int. Conf. Artificial Immune Systems (ICARIS)*, Canterbury, U.K., 2002, pp. 3–11. Session I.

[12] H. Y. K. Lau and V. W. K. Wong, "Immunologic control framework for automated material handling," in *Proc. Int. Conf. Artificial Immune Systems (ICARIS)*, Edinburgh, U.K., 2003, pp. 57–68.

[13] ——, "A strategic behavioral-based intelligent transport system with artificial immune system," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, The Hague, The Netherlands, 2004, pp. 3090–3914.

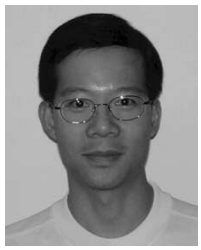
[14] J. M. Cruse and R. E. Lewis, *Atlas of Immunology*. Boca Raton, FL: CRC Press, 1999.

[15] K. D. Elgert, *Immunology: Understanding the Immune System*. New York: Wiley, 1996.

[16] S. A. Hofmeyr and S. Forrest, "Immunity by design: An artificial immune system," in *Proc. Genetic and Evolutionary Computation Conf.*, San Francisco, CA, 1999, vol. 2, pp. 1289–1296.

[17] D. J. Smith, S. Forrest, and A. S. Perelson, "Immunological memory is associative," in *Artificial Immune Systems and Their Applications*, D. Dasgupta, Ed. Berlin, Germany: Springer-Verlag, 1998, pp. 105–112.

[18] J. Kim and P. Bentley, "Immune memory in the dynamic clonal selection algorithm," in *Proc. Int. Conf. Artificial Immune Systems (ICARIS)*, Edinburgh, U.K., 2003, pp. 59–67. Session II.
[19] O. Beltran and F. Nino, "A change detection software agent based on immune mixed selection," *Evol. Comput.*, vol. 1, no. 12–17, pp. 693–698, May 2002.
[20] M. Ayara, J. Timmis, R. de Lemos, L. de Castro, and R. Duncan, "Negative selection: How to generate detectors," in *Proc. Int. Conf. Artificial Immune Systems (ICARIS)*, Edinburgh, U.K., 2003, pp. 89–98. Session III.
[21] D. W. Lee and K. B. Sim, "Artificial immune network-based cooperative control in collective autonomous mobile robots," in *Proc. IEEE Int. Workshop Robot and Human Communication*, Sendai, Japan, 1997, pp. 58–63.
[22] H. Meshref and H. VanLandingham, "Artificial immune systems: Application to autonomous agents," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, Nashville, TN, 2000, vol. 1, pp. 61–66.
[23] R. Michelan and F. J. Von Zuben, "Decentralized control system for autonomous navigation based on an evolved artificial immune network," in *Proc. Congr. Evolutionary Computation (CEC)*, Honolulu, HI, 2002, vol. 2, pp. 1021–1026.
[24] A. Freitas and J. Timmis, "Revisiting the foundations of artificial immune systems: A problem-oriented perspective," in *Proc. Int. Conf. Artificial Immune Systems (ICARIS)*, Edinburgh, U.K., 2003, pp. 229–241.
[25] L. N. de Castro and J. Timmis, *Artificial Immune System: A New Computation Intelligence Approach*. Berlin, Germany: Springer-Verlag, 2002.
[26] A. K. Abbas and A. H. Lichtman, *Basic Immunology: Functions and Disorders of the Immune System*. Philadelphia, PA: Saunders, 2001.
[27] L. J. Eales, *Immunology for Life Scientists: A Basic Introduction: A Student Learning Approach*. Chichester, U.K.: Wiley, 1997.
[28] C. Coello, D. Rivera, and N. Cortes, "Use of an artificial immune system for job shop scheduling," in *Proc. Int. Conf. Artificial Immune Systems (ICARIS)*, Edinburgh, U.K., 2003, pp. 1–10.
[29] E. Bendiab, S. Meshoul, and M. Batouche, "An artificial immune system for multimodality image alignment," in *Proc. Int. Conf. Artificial Immune Systems (ICARIS)*, Edinburgh, U.K., 2003, pp. 11–21.
[30] L. Goncharova, Y. Melnikov, and A. Tarakanov, "Biomolecular immunocomputing," in *Proc. Int. Conf. Artificial Immune Systems (ICARIS)*, Edinburgh, U.K., 2003, pp. 102–110.
[31] F. Esponda, E. Ackley, S. Forrest, and P. Helman, "Online negative database," in *Proc. Int. Conf. Artificial Immune Systems (ICARIS)*, Catania, Italy, 2004, pp. 175–188.
[32] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Boston, MA: McGraw-Hill, 2001.
[33] I. M. Roitt, J. Brostoff, and D. Male, *Case Studies in Immunology: Companion to Immunology*, 5th ed. London, U.K.: Mosby, 1998.



Henry Y. K. Lau (M'98) received the B.A. degree in engineering science and the D.Phil. degree in robotics from the University of Oxford, Oxford, U.K., in 1991.

Prior to joining The University of Hong Kong, Hong Kong, as an Associate Professor, he was a College Lecturer at Brasenose College, Oxford, and has been working in the industry as a Senior System Engineer with AEA Technology plc., U.K., working on projects involving system integration and telerobotics systems for the nuclear industry. His research interest includes intelligent automation, automated warehousing, interactive virtual reality, and design and evaluation of automated material-handling systems, such as automated warehouses using simulation and virtual-reality techniques.

Vicky W. K. Wong received the B.Eng. degree in manufacturing systems engineering with management and the M.Sc. degree in information technology from the University of London, London, U.K., in 2001 and 2002, respectively. She is currently working towards the Ph.D. degree at the Department of Industrial and Manufacturing Systems Engineering, University of Hong Kong, Hong Kong.

Her research interests include artificial immune system, multiagent systems, autonomous agents, and behavior-based systems.