Soft Comput (2019) 23:1199–1218
https://doi.org/10.1007/s00500-017-2839-5

METHODOLOGIES AND APPLICATION

Distributed task allocation in multi-agent environments using cellular learning automata

Maryam Khani[1] · Ali Ahmadi[2] · Hajar Hajary[3]

Published online: 4 October 2017

Abstract People's safety is threatened by the repetition of critical events. Many people lose their lives due to unprofessional rescue operation as well as time pressure of the rescue operation. A key problem in urban search and rescue teams, considering the severe turbulence and complexity of the environments which are hit by a crisis, is the coordination between the team members. In order to solve this problem, an effective plan would be the provision of measures where human works with intelligent assistant agents to assign the tasks in any way. Dynamic tasks are identified by the human agent of the rescue team in the crisis environment and are characterized by spatial–temporal characteristics assigned to the appropriate rescue team by the intelligent assistant agents who apply intelligent decision-making techniques. The objective of this study is to propose a new approach for allocating spatial–temporal tasks in multi-agent systems through cellular learning automata as the decision-making technique. Results obtained here indicate that this proposed model can significantly improve the rescue time and space. Rescue teams could cover all critical areas by going through the minimum distance to make maximum use of time.

Keywords Spatial–temporal task allocation · Multi-agent system · Cellular learning automata · Earthquake emergency response · Geospatial simulation · Urban search and rescue

1 Introduction

The occurrence of artificial crisis and natural disasters impose great losses on human life in all aspect. Immediate reactions to these events are necessary to protect the lives and properties of people at the time of the incident and to reduce life loss. These reactions require immediate operations and plans to coordinate forces, facilities and resources (Khalil et al. 2009, 2008; Ashish et al. 2007; Jinguo et al. 2007; Massaguer et al. 2006; Nourjou et al. 2014b). The objective here is to find answer to the main question of this study, that is, 'How can we obtain a more efficient manner to optimize the coordination of human teams in performing the assigned tasks with respect to time in a dynamic environment with spatial characteristics at the time of critical events?' In the beginning of the disaster occurrence, assigning clear tasks to rescue teams is difficult. Therefore, it is necessary to first identify the environment and then to assign the tasks thereupon in a dynamic manner. In this context, the artificial intelligence technology offers multi-agent systems as a solution for designing programs which necessarily have decentralized control and dynamic environments (Russell and Norvig 1995).

A multi-agent system (MAS) is a system composed of multiple interacting intelligent agents within an environment. The agents here could equally be robots, humans and their combinations. MASs can be applied to solve problems which are difficult or impossible for an individual agent to solve as

Communicated by V. Loia.

✉ Maryam Khani
maryam.khani@hdr.mq.edu.au

Ali Ahmadi
ahmadi@kntu.ac.ir

Hajar Hajary
hajar_hajary@yahoo.com

1 Department of Computing, Macquarie University, North Ryde, NSW 2109, Sydney, Australia

2 Department of Computer Engineering, K.N. Toosi University of Technology, Tehran, Iran

3 Department of Computer Science and Research Branch, Islamic Azad University, Tehran, Iran

a response to crisis. In MAS, agent characteristics consist of being autonomous, having a local view of environment and capable of learning, planning, coordinating and decentralized decision making.

Here exist several MASs in managing the hazardous events and simulating emergency responses. Some of these multi-agent systems include (Massaguer et al. 2006; Russell and Norvig 1995; Schurr et al. 2009; Schoenharl and Madey 2006; Adams et al. 2008; Jain and McLean 2004; Murakami et al. 2002): DrillSim, ALADDIN, RoboCup Rescue, FireGrid, Wiper, and DEFACTO. The key feature in the responses based on MASs is the use of assistant agents of critical operators able to interact with one another. The agents help the critical operators in planning, decision making, communicating with the other agents, and managing the information (Schurr et al. 2009; Adams et al. 2008; Jain and McLean 2004; Murakami et al. 2002; Maheswaran et al. 2010; Nourjou et al. 2011; Nourjou and Hatayama 2011; Nourjou et al. 2014c, a).

According to the many studies run on MASs, design of algorithms based on the best allocation of task is a competitive issue for which several methods for task allocation are proposed (Dias and Stentz 2000; Song et al. 2009; Gerkey et al. 2003; Ferreira et al. 2009; Koes et al. 2005; Rasekh and Vafaeinezhad 2012; Suarez Baron 2010; Hunsberger and Grosz 2000; Ham and Agha 2007).

One of the important coordination techniques in task allocation is the contract net protocol (Ferreira et al. 2009) which consists of four steps: (1) problem diagnosis, (2) tasks announcement, (3) bidding, and (4) awarding. For example, Dias and et al. have introduced methods based on auction and bidding (Dias and Stentz 2000; Dias 2004; Dias et al. 2006) and Song et al. (2009) have presented a Distributed Bidirectional Auction algorithm for multi-robot systems coordination. Moreover, other forms of tasks assignments are developed with regard to marketing by employing auction and bidding (Hunsberger and Grosz 2000; Ham and Agha 2007; Dorigo 2005; Dorigo et al. 2013; Ham and Agha 2008; Gerkey and Mataric 2004; Liekna et al. 2012; Hussein and Khamis 2013).

Auction and bidding methods are market-based mechanisms. Items on sale are tasks, and bidders are the agents. Agents offer suggestions to an auctioneer according to their abilities and resources. After receiving all the offers, the auctioneer allocates the tasks between the bidders. In this method, offers are transferred through contract network protocol (CNP) among the agents.

A proposed model based on a decentralized approach and the basis of multi-agent systems is presented by Nourjou et al. (2011) and Nourjou and Hatayama (2011). The design of this proposed model is in accordance with the distributed tasks allocation algorithm based on auction and bidding method. In the task allocation method based on auction and bidding method, communication channels are always heavily loaded. When an agent observes a task, it follows a protocol of messages to make a contact with the other agents which are potentially able to do the task. Making this connection is very time consuming and limits the application of auction and bidding method for allocating tasks among the agents (Dos Santos and Bazzan 2011). Rahwan and Jennings (2007) presented a method for coalition and cooperation among agents. Despite using a new method for reducing the amount of communication between agents, an agent still needs to be aware of other agents' capabilities in order to calculate the value of a coalition. Shiroma and Campos (2009) also proposed a framework for the coordination and distribution of tasks among a set of heterogeneous robots named CoMutaR (coalition formation based on multitasking robot). This method is based on auction and bidding to form coalitions concurrent through actions and follows an auction process. Scerri et al. (2005) proposed an approximation algorithm named low-communication approximate DCOP (LA-DCOP) where a token-based protocol is applied.

A distributed constraint optimization problem (DCOP) consists of a set of variables that can assume values in a discrete domain (Grinshpoun and Grubshtein 2013; Leite et al. 2014). Each variable is assigned to an agent who has control on its values. The duty of agents is to select values for the variables, that is, optimizing a global objective function. Task allocation problem can be returned to a DCOP which considers the agents as the variables and tasks as the domain of values. Agents identify a task in the environment and create a token to do the task and vice versa. Allowing the agent to decide whether or not to do a task based on its capabilities and on a threshold (therefore, it keeps the token for itself or sends it to another agent) at which it applies complex maximization functions. Division of labor in many species of social insects has long been the subject of numerous studies (Dos Santos and Bazzan 2011; Theraulaz et al. 1998; Yang et al. 2009; Quiñonez et al. 2011a; Yasuda et al. 2014; Brutschy and Pini 2014; Cornejo and Dornhaus 2014). A complex colony behavior emerges from individual workers. One of the most outstanding aspects of labor division is plasticity, which allows them to perform the different tasks that maintain the colony's viability.

Such methods are based on individuals' internal response threshold, and they are associated with tasks stimulant. It is assumed that j tasks are available and for each task $j \in J$, a s_j stimulant is appointed. A set of I individuals who can perform the tasks of J set is given. Every individual $i \in I$ has a response threshold θij, related to the probability of reacting to the stimulus associated with task j. A response threshold θ is an internal variable that determines the tendency of an individual to response to the stimulus s and perform the associated task. Threshold θ has the probability to response to $s < \theta$, that is low and for $s > \theta$ that is high. In this model, the individual internal threshold θ_{ij} and the task stimulus s_j applied to compute the probability (tendency) $T_{ij}(s_j)$ of the individual$i$ to perform task j, as shown in formula (1):

$$T_{ij}\left(S_j\right) = \frac{S_j^n}{S_j^n + \theta_{ij}^n} \quad (n > 1) \tag{1}$$

where $n > 1$ is the steepness of the threshold. Each individual can do every task if the stimulus corresponding to the task is high enough to overcome the individual's internal threshold. However, some of the simple methods presented in this field apply fixed thresholds to drive different behaviors, Fathy Navid and Aghababa (2013), while researchers mostly adopt methods where the thresholds are adapted over time, such as Labella et al. (2006), Campo and Dorigo (2007), and Liu et al. (2007), Castelloh et al. (2013). A common formalization of adaptive thresholds that they are named reinforced response is proposed by Ferreira et al. (2008) and Ikemoto et al. (2010).

Task allocation techniques and methods are often based on "auction and bidding" which are based on CNP, methods based on threshold response and token-based methods. Ferreira et al. (2008) compare a reinforced response threshold method with a token-based method in a RoboCup rescue scenario; Kalra and Martinoli (2006) compare auction-based methods and token-based approaches. A comparison between these methods indicates that auctions require a lot of communication, but when the task information is accurate, auction-based methods are run in a better way. On the contrary, when the task information is not accurate, the threshold-based methods are performed as well as the auction-based methods with a fraction of the expense in communication. For example, the optimized consensus-based bundle algorithm (CBBA) is proposed for spatial–temporal task allocation. This algorithm makes interactions among time, place, and communications (Godoy and Gini 2013). This method is based on auction and bidding and still has the problem of demanding sizable of communications.

In these methods, in the design of the structure of agents, the ability to learn and be aware of the decisions of other teams are not of concern, and rescue teams are assigned to their tasks regardless of the result of the search teams' choice. Consequently, the best choice will not be made for all search teams to assign their identified tasks to rescue teams. The concept of interaction and teamwork is not defined for the members of the rescue teams, which is a drawback in this procedure.

Decentralized solution where human teams with assistant of their intelligent agents in an individual and local manner select an appropriate team and assign their rescue tasks while all recognized rescue tasks are optimally distributed is the focused here. In this context, the intelligent assistant agents erupted with learning ability instead of a centralized controller is applied. The main objective of this article is minimizing the rescue time. To accomplish this, the intelligent assistant agents of search teams must have flexibility in making decision, That is, they should be able to recognize new rescue tasks and assign them to rescue teams optimally while coordinated with other agents. Another important point is the dynamic features of the critical environments while new rescue tasks are recognized during the operation, and decisions must be made with very little knowledge regarding the given environment.

So if the search team assigns a known task to a rescue team through a definite model, in reality it has not considered the choices of other search teams (among the available ones) and has not applied the learning method (due to adopting definite model). For example, search team A has identified a rescue team at the least distance from the critical point and assigns the task to this team, while it is possible that this selected rescue team is a better selection for other search teams such as B, C, and D which might were located in nearest to this selected rescue team. This implies that search team A has assigned a task to rescue team only by considering the fixed distance parameter which in turn leads to a delay in rescue performance for another search team, thus, a wrong priority. Accordingly, it is necessary for the search team to assess the possible outcomes of any solution and assignment based on discussion made by other search teams and gives the final response. Thus, there exist more critical emergency cases in need of immediate rescue, while due to wrong prioritization at the given time interval some rescue teams not fit for that certain condition are assigned for the task. This wrong prioritization, with respect to time needed to rescue of injured persons' life, is of major concern, that is, a matter of life and death. In a deterministic method, tasks are allocated among rescuers without optimizing the time factor and the space because in deterministic methods learning methods are not applied. By adopting probabilistic methods, the results of selected action are investigated (result is the reward or penalty) and this learning process continues until the appropriate result is obtained.

Statistical methods are adopted to solve the main conflicts between the mobile robots by proposing some new and innovative solutions for important problems such as navigating, locating, tracking, and controlling of the robots. These methods are adopted to solve coordination problems among several robots in order to generate an auto-selection among the specified heterogeneous tasks (Quiñonez et al. 2011b). In this article to achieve the mentioned objectives, the stochastic reinforcement-learning algorithms is applied in designing intelligent assistant agents.

The agents act in complex, large-scale, dynamic, and unpredictable environment and know little about each other due to the information distribution; and due to the dynamic nature of the environment, the decisions made by other agents may change over time. In order to make the best decisions, an agent must not only know the consequences of its own decisions, but also learn how to coordinate its decisions with those of the other agents. Thus, the use of reinforcement-learning methods, especially when the agents have little knowledge of the environment, leads to more optimal results.

Attempt is made in this article to propose a new approach for allocating spatial–temporal tasks through a stochastic reinforcement-learning algorithm based on cellular learning automata (CLA) where agents with knowledge of the decision of other agents and by coordination with each other can draw the best decisions for task allocation and assist their human agents in drawing the best decisions. The rescue time and space, two vital parameters in disaster crisis, are improved here. All rescue teams could cover all critical areas by covering the minimum distance to make maximum use of time.

Rescue teams have the software agents equipped with a learning automaton. The CLA is adopted to model interactions and contacts among the search agents. In this sense, the search team is an autonomous entity that performs its search tasks and through its intelligent software agent, records the new identified tasks, and delivers them to the most appropriate rescue team available in the operating area.

In this article, a distributed approach is applied since the focus is an autonomous and decentralized technique, where agents themselves are responsible for assigning tasks. The proposed model is evaluated in accordance with two deterministic models: (1) the auction and binding and (2) response threshold based. The result indicates that agents have the ability to select an appropriate rescue team for assigning their rescue tasks in autonomous and individual manner without any need to take control of the central controller. Accordingly, time which is important factor could improve by adding learning ability and using probability algorithms.

This article is organized as follows: Description of this proposed model is presented in Sect. 2. The cellular learning automata-based model is introduced in the Sect. 3. The implementation of a geospatial simulation environment for spatially distributed intelligent assistant agents using Visual Studio, software on .Net framework and C# programming language, is described in Sect. 4; simulation results and discussions are provided in Sects. 5 and 6.

2 Definitions

2.1 Statement of the problem

During a crisis or disaster, the search and rescue teams are faced with a complicated and unpredictable environment where the first 72 h of search and rescue operations are of vital importance. In such circumstances, considering the time constraints and the magnitude and complexity of the incident, multiple-organized operations in large scales for crisis management are of essence. An effective plan for such an organization is one which enables the humans to cooperate with the intelligent agents as assistants in order to in an efficient manner (Nourjou et al. 2011; Nourjou and Hatayama 2011).

The spatial–temporal coordination problem (STCP) is represented by a set of K tasks, denoted as $L = \{l_1, l_2, \ldots, l_K\}$ with temporal (specifies the time limit to run a task as a deadline) and spatial (indicates the location coordinates of the task) constraints; these tasks have to be dynamically identified by a set of N search teams that is represented by $S = \{S_1, S_2, \ldots, S_N\}$ which is delivered into a set of M rescue teams represented by $R = \{R_1, R_2, \ldots, R_M\}$ in the same operational areas. This process should be run in a manner that the best choices are made for all the search teams in assigning their identified tasks. Best choice here is selecting the best rescue team that is the team with the lowest cost at the lowest distance from a critical point with the highest ability, and the most important parameter is that the selected rescue team is not a vital selection for other rescue teams. The objective of solving this problem is in reducing rescue time so that more people could save because time plays significant role in rescue operations.

Selecting the appropriate rescue team means choosing a team that is better than the others in terms of selection parameters. For example, the critical point a, with the search team S_i which is identified as the rescue task L_i, and the critical point b with the search team S_j which is identified as the rescue task L_j, are taken into account. If rescue team R_i is the closest to a critical point a, and if this rescue team is not a vital selection for search team $S_{j,}$ it will be an optimal option for search team $S_{i.}$. If the search team S_j by losing this rescue team is obliged to choose another rescue team, which may be located at a farther distance and lose more time, while the search team S_i could choose another rescue team and lose less time, this R_i rescue team is a vital selection for search team Sj and therefore is not an optimal choice for the search team S_i.

By adding learning ability to assistant agents of search teams in selecting rescue team for assigning rescue task, in addition to the factor such as distance of rescue team to critical point in performing rescue task, the ability of rescue team is the other important factor to be assessed. Each agent of search team by considering the decision of other search teams for assigning its rescue tasks to rescue teams has important effect on declining of rescue time. Here it is deduced that this effect is visible compared to other methods. The objectives of this article are the presentation of cooperation of human and its assistant agents for solving STCP and adding learning ability to assist agent while cellular learning automata is proposed to accomplished this with respect to its advantages.

The use of rescue time and space, the two important parameters in disaster crisis, is improved here. So all rescue teams could cover all critical areas in the minimum distance and maximum use of time. The main objective of such an approach is to increase and maximize the number of the rescued victims in the shortest possible time.

A list of different field teams in the search and rescue operations in the event of crisis is tabulated in Table 1. In this article, the optimal method for creating coordination among search and rescue teams is noted.

2.2 Properties of this proposed approach

Generally speaking, the multi-agent system which is designed for crisis managing should cover response domain requirements including: filtering and data fusion methods, decision-making and machine learning methods, managing the interaction among multiple actors by designing interaction mechanism methods, expanded studies on system architecture and information exchange topologies (Khalil et al. 2009, 2008). The specifications of the proposed multi-agent system that cover all crisis response domain requirements for solving STCP are categorized as follows:

1. Multi-agent system specifications
This proposed system is designed based on multi-agent system architecture, communication and collaborative among agents, agent diversity (our assumption in this article the focus is on two types of teams: search and rescue teams).

2. Cooperation the human and intelligent assistant agent
In real situation, any human of each team is equipped with a personal digital assistant (PDA) device which runs a personal intelligent assistant agent and GPS map. A personal intelligent assistant agent is a software agent that assists its user (human) in drawing decision, coordination, communication with other assistant agents, managing the information, etc. On the GIS map provided by intelligent assistant agent, human finds and locates the critical point and doing the search, inserting new related information in local database and defining rescue task for this point are assigned to him, and then, he/she asks his agent to update central database and allocate rescue task of this critical point to the best rescue team.

3. Data management
Data management is run through a central database and distributed (local) databases, supporting GIS data. The two types of data management: the central database and distributed (local) databases are considered. Each intelligent assistant agent works with its own local database which is embedded in its PDA. All intelligent assistant agents have access to the central database for data sharing improvement. In real situation, the central database is a GIS server or Google maps (Jain and McLean 2004).

4. Coordination
Intelligent assistant agents are responsible for task allocation, cooperation in improving the rescue time, reduce deaths and injuries, and collaborative decision making.

5. Learning
As an important character of the proposed approach, each intelligent assistant agent is ability to learn to enhance its decision making for task allocation and ability of agents monitoring.

6. Others specification
Adapting the methods based on extreme teams (Dos Santos and Bazzan 2011) makes the specification of the proposed model flexible.

The proposed method contains all of these characteristics.

2.3 Cellular learning automata

Research in learning automata (LA) began with Testlin (1961) who introduced the application of deterministic and stochastic automata operations in a random environment as a learning model. The term "learning automata" was first published in a survey article by (Narendra and Thathachar 1989). The objective of LA is to 'determine the optimal action out of a set of allowable actions, where the optimal action is defined as the action that maximizes the probability of being rewarded.' Learning automata (LA) is a sophisticated reinforcement-learning model for decision making in stochastic and unknown environments (Narendra and Thathachar 1989; Lakshmivarahan 1981; Najim and Poznyak 1994). An automaton can select an action among a set of actions as its output. Once the action is selected and executed, it is evaluated by the environment and the corresponding feedback is sent to the learning automata either as a positive feedback signal (i.e., in case the action is done properly) or a negative one (i.e., in case the action is done improperly). The value of this signal determines which actions should be chosen for the following steps. This process makes the automata to gradually converge to the most appropriate action regarding the environmental criteria. However, LA is capable of learning the optimal action, among a set of finite actions, by repeating the two-step process of: (i) at each reiteration, the learning automaton chooses one of its actions based on its selection probability and performs it on the environment, and (ii) the automaton receives a reinforcement signal from the environment and modifies its behavior accordingly. The closed-loop interaction between a stochastic automaton and the random environment is shown in Fig. 1.

The learning automata are of two classes of: variable structure stochastic automata (VSSA) and fixed structure stochastic automata (FSSA) (Narendra and Thathachar 1989). Transitions of fixed structure stochastic automata are determined by state transition probabilities that are fixed with

Table 1 Describing different field teams and their tasks in the search and rescue operation (Nourjou et al. 2011; Nourjou and Hatayama 2011)

Teams	Tasks
Incident commander (IC)	Central commander and controller and the coordinator of other teams
Loss estimation teams	To estimate the scale of damage
Search team	Search critical locations and collecting actual data from damage scale
Rescue team	Rescue the critical locations with performing the set of tasks assigned to it
Medical team	Provide the medical equipment
Firefighting team	Control the chances of fire hazards and extinguishing fires
Traffic Police team	Control routes and opening the blocked routes
Logistics team	Provide logistics and food supplies

time. The FSSA undergoes a slow convergence speed in comparison with VSSA. In this article, an automaton is applied with variable structure. Variable structure stochastic automata is represented by a quadruple $\{\alpha, \beta, P, T\}$ where $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ which is the set of actions of the automaton, $\beta = \{\beta_1, \beta_2, \ldots, \beta_m\}$ is its set of inputs, $P = \{p_1, \ldots, p_r\}$ is the probability vector for selection of each action, and $P(n+1) = T[\alpha(n), \beta(n), P(n)]$ as the learning algorithm while the function of T is the reinforcement algorithm, which modifies the probability vector P with respect to the performed action and received response (Narendra and Thathachar 1989). According to the variable structure learning automata, the following equations are calculated to obtain the probability vector for the appropriate and inappropriate responses from the environment (Fathy Navid and Aghababa 2013; Narendra and Thathachar 1989; Beigy and Meybodi 2004):

1. The environment appropriate response ($\beta = 0$):

$$\begin{aligned} P_i(t+1) &= P_i(t) + a(1 - P_i(t)) \\ P_j(t+1) &= (1-a)P_j(t) \qquad \forall j \quad j \neq i \end{aligned} \tag{2}$$

2. The environment inappropriate response ($\beta = 1$):

$$\begin{aligned} P_i(t+1) &= (1-b)P_i(t) \\ P_j(t+1) &= \left(\frac{b}{r-1}\right) + (1-b)P_j(t) \qquad \forall j \quad j \neq i \end{aligned} \tag{3}$$

In Eqs. (2) and (3), $P_i(t)$ is the probability of selecting action α_i at time t, 'a' is the reward parameter, 'b' is the penalty parameter, and 'r' is the number of possible actions. When $a = b$, the automaton is named L_{RP}. If $b = 0$, the automaton is named L_{RI} and if $0 < b << a < 1$, automaton is named L_{ReP}.

The correlation between the learning automata and the environment is shown in Fig. 1. In this type of automata, if α_i act is selected at nth stage, and it receives a favorable

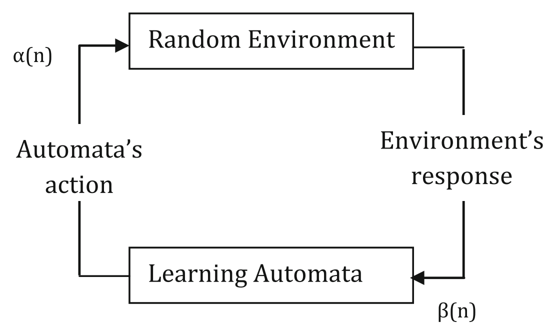


Fig. 1 Closed-loop interaction between a learning automaton and environment

response from the environment, the probability of $P_i(n)$ will increase and other probabilities decrease. For undesirable response, $P_i(n)$ probability reduces, and other probabilities increase. However, changes are made in a way that the sum of $P_i(n)$ always remains constant and is equal to one. This interaction between LA and the environment can guide the LA toward selecting the optimal action.

LAs are successfully used in many applications such as intrusion detection in sensor networks (Misra et al. 2009), database systems (Fayyoumi and Oommen 2009), solving shortest path problem in stochastic networks (Misra and Oommen 2009), assigning channel in wireless sensor networks (Moghiss et al. 2010), managing traffic signals (Barzegar et al. 2011), and ranking function discovery algorithm (Akbari Torkestani 2012).

Cellular automata introduced by Von Neumann 1996 are mathematical models for defining systems that consist of a large number of simple identical components with local interactions. Cellular automata (CA) is an abstract dynamical system consisting of a large number of identical simple cells that are distributed in a grid-like structure and can produce complex phenomena (Fathy Navid and Aghababa 2013). Each CA can be identified with a five-tuple $\{\Phi, \Delta, s_i^t, \phi, T\}$, where Φ is a set of cells which are arranged in some regular forms such as grid, Δ is the set of finite states, s_i^t denotes the state of i-th cell at t-th time step, ϕ is a set of cells surrounding a given cell, and $T : \left(s_i^t, \phi\right) \rightarrow \Delta$ is the transition function which is used to determine the next state of a cell according to its current state and the states of the cells in its neighborhood.

CLA is a mathematical model for simulating dynamical complex systems that include large number of simple components. These simple components have learning capabilities and act together to produce complex behavioral patterns. In other words, a CLA is a cellular automaton in which a learning automaton is assigned to its every cell (Beigy and Meybodi 2004). The learning automaton residing inside each cell determines the state of the cell on the basis of its action probability set. The active rule in CLA and the actions selected by the neighboring cells determine the reinforcement signal to the learning automata residing in that cell. The neighboring learning automata of any cell constitutes its local environment. The state of the cell is determined by the action probability set of the learning automaton residing in that cell. The initial value of the state may be set based on the past experience or randomly. After initializing the states, the reinforcement signal to each learning automaton is determined by the CLA rule. Then, each learning automaton updates its action probability set based on the reinforcement signal and the chosen action. This process continues until the desired result is obtained (Beigy and Meybodi 2004; Fathy Navid and Aghababa 2013).

Formally, a d-dimensional cellular learning automata can be defined as $A = (Z_d, \Phi, A, N, F)$, where Z_d is a lattice of d-tuple of integer numbers, Φ is a finite set of states, A is the set of learning automatons each of which is assigned to each cell of the cellular automata, $N = \{X_1, X_2, \ldots, X_m\}$ is a finite subset of Zd named neighborhood vector where m represents the number of neighboring cells and $X_i \in Z_d$ and finally F is a set of action functions each of which determines the next action of each automaton. The neighborhood vector determines the relative position of the neighboring cells from any given cell u in the lattice Zd. The neighbors of a particular cell u are set of cells which are located in a neighborhood radius r. We assume that there exists a neighborhood function $N(u)$ mapping a cell u to the set of its neighbors.

The CLAs are used in many applications (Fathy Navid and Aghababa 2013) such as image processing (Meybodi and Kharazmi 2004), channel assignment in cellular networks (Beigy and Meybodi 2003), call admission control in cellular networks (Beigy and Meybodi 2008) and sensor networks (Esnaashari and Meybodi 2008), dynamic point coverage problem in wireless sensor networks (Esnaashari and Meybodi 2010), and hybrid web recommender system (Talabeigi et al. 2010).

Cellular learning automata can be viewed as a simple model of a spatially extended decentralized system made up of a number of individual components (cells). The communication among constituent cells is limited to local interaction. Each individual cell is in a specific state which changes over time depending on the states of its local neighbors. The overall structure can be viewed as a parallel processing device. However, this simple structure when iterated several times produces complex patterns displaying the potential to simulate different sophisticated natural phenomena. Thus, cellular learning automata can be a powerful tool to model the interactions among learners.

2.4 CLA in the proposed model

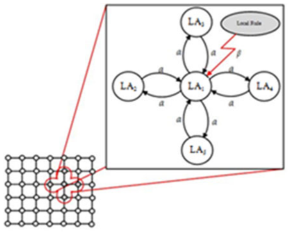
With respect to the problems addressed in Sect. 1.2, there exists a set of agents (searched rescue) which are in close contact with one another in a special environment. In order to solve the STCP problem in the best manner, the following two measures should be of concern:

1. The need to increase the learning capacity of each agent
2. Providing communication among all agents through which they will be aware of one another's discussion, thus, low overhead

By implementing these measures, the sets of stronger interconnected learners with low overhead would be yield. The CLA method with all its characteristics explained in Sect. 2.3 can be the best choice in meeting these requirements. Accordingly, any agent equipped with learning power is considered as a LA. Since the agents are located in spatial environment where discussion making of each is considered by the other for having more rescue in the least time, they behave like neighboring cells in CLA, thus, the reason to adopt CLA in modeling of these agents. The CLA is chosen here since it is appropriate for the factors necessary in solving the problems. By comparing the results here, obtaining optimized responses become possible.

So, we have equipped the software agents of the search team with a learning automaton. Also, we have created CLA to model interactions and contacts among search teams while each cell is equal to each search team. Thus, the search team is an autonomous entity that performs its search tasks and, by using its intelligent software agent, records the new identified tasks and delivers them to the most appropriate rescue team present in the operating area.

When an earthquake occurs, the crisis region is divided into several areas named an operational area. A number of search teams and rescue teams are deployed to each area of operation. In the proposed model, search teams in each operational area are defined as neighbors. The search teams are identified as the cells of the CA based on their operational regions. In order to coordinate the operations of the agents in the environment, the concept of cellular learning automata is applied in the proposed model to share the information



α	action of each learning automaton
β	Input of each learning automata
LA_i	Learning automata with index i={1,2,3,4,5}
L_k	Task with index k={1,2,3,4,5}
R_m	Rescue team with index m={1,2,3,4,5}

(a)

LA_1	Assign l_1 to R_2
LA_2	Assign l_2 to R_2
LA_3	Assign l_3 to R_3
LA_4	Assign l_4 to R_3
LA_5	Assign l_5 to R_5

After n Iteration

(b)

LA_1	Assign l_1 to R_2
LA_2	Assign l_2 to R_3
LA_3	Assign l_3 to R_5
LA_4	Assign l_4 to R_1
LA_5	Assign l_5 to R_4

Fig. 2 A schematic of using cellular learning automata for task allocation among rescue teams. **a** First task allocation (decision of each automaton). **b** Task allocation after learning (decision of each automaton)

among the agents working in the same operational area. The main feature of CLA in this article is that learning automata in each cell is executed in parallel and all assistant agents follow the same rules.

A variable structure stochastic automaton with L_{ReP} (linear Reward-e-Penalty) scheme is applied for each cell (in design of search team's intelligent assistant agent). The linear Reward-e-Penalty scheme is a simple linear model, while it is an optimal model (Narendra and Thathachar 1989). The degree of freedom L_{ReP} is greater than the linear model L_{R-I}, because L_{ReP} has two parameters a and b. In addition, this model is simpler than nonlinear models. By adopting this method, the desired convergence takes place in a rapid manner. Each automaton is responsible for assigning an appropriate rescue team to a certain task.

A schematic of using CLA for allocation of tasks among search teams is shown Fig. 2. The details related to a learning automaton LA_1 are shown separately as an instance. For example, the table for allocating five tasks to five rescue teams by five learning automatons before and after learning is shown. In the first decision making, automatons have not distributed the allocation of tasks among rescues teams in an appropriate manner; however, after a number of 'n' repetitions of learning, they become able to optimize the distribution of all tasks among the rescue teams.

3 Allocating spatial–temporal tasks by using CLA

3.1 Different stages of distributed "spatial–temporal" task allocation

The series of steps for the task allocation are described in 9 different stages in detail, and Fig. 3 shows the inner-connection between these steps by activity diagram among.

1. At the beginning of the process, the information related to the location of critical places and predicted numbers of victims is collected by the loss estimator team, and they are inserted in database and located on the map. The purpose of these teams is to locate damaged buildings, estimate number of potential victims, collapse debris, and then, report situation information to the IC. Afterward, identified by $S = \{S_1, S_2, \ldots, S_N\}$ in each operation area are deployed to the locations which are more critical than other places in the operation area with command of IC. In other words, at the beginning of the process command center selects the most critical locations in each operation area based on collected information and sends messages to the search teams of those operation areas.
2. Now, in each operation area, the search teams move to their assigned critical places to do search tasks and collect more information such as the actual amount of damages, the actual number of victims, the level of relief needed, time limit for the required relief, the danger of catching fire, collapse debris.
3. Then, they edit the estimated information; enter the coordinates of unexpected critical places as the new data in the local database on the PDA. They ask their personal intelligent assistant agents to enter all updated information in the central database. This information will be shared with all the teams involved in the critical incident and will help them to make the best decisions. Also, human search team asks its personal intelligent assistant agents to allocate rescue tasks of these points to the best rescue teams. Personal intelligent assistant agents assist the human agents of search in making decision for allocating the identified relief tasks to the appropriate rescue team. Now there is a set of tasks shown by $L = \{l_1, l_2, \ldots, l_K\}$. The search team members also use their personal experience to estimate the level of relief needed for each place.
4. The assistant agents of all search teams could be informed of the precise information relevant to the critical places through connecting to the local and the central database and using the locations map. At this stage, at the time of allocating the identified relief tasks to the appropriate rescue team, assistant agent of the search team selects its appropriate rescue team from a set of rescue teams that are identified by $R = \{R_1, R_2, \ldots, R_M\}$, according to the new entered data as well as the important parameters (Nourjou et al. 2011; Nourjou and Hatayama 2011). At the beginning of the operation, the automaton determined the selection of probability for each rescue team based on the information of the existing rescue teams in the operational area as well as the new information obtained from the critical points. The automata considered different parameters for calculating the initial selection probability of each rescue team such as the distance between the rescue team and the critical point, the time required for

relief, the rescue level of critical point (shows level of rescue at critical point that depends on the amount of damage at that point, the number of injured people, the level of injuries, the possibility of a fire, power cuts, etc.), the critical point status, the rescue team's abilities (including all abilities of the rescue team for relief: the number of rescue teams, the equipment, team energy with respect to the number of tasks that have been done, the skill level of the team), the number of unfinished tasks assigned to the rescue teams. Therefore, the initial selection probabilities of different rescue teams are not equal. Algorithm (1) was used for this purpose. In this algorithm, the calculation method of possibilities' list related to rescue teams settled in the crisis area is shown by assistant agent of each search team.

The formulas of algorithm were achieved based on (Vafaeinezhad et al. 2009), which were obtained by empirical tests and meeting with specialist rescue teams.

The Algorithm (1) for calculating the probability for each rescue team by assistant agent of each the searcher team

```
if search operation in critical point = finished then
    rescues= select rescues in operation area;
    listProbabilities = NewList();
    foreach rescue in rescues
        distance = Getdistance();
        cost ={(distance/5) + (ability of rescue team) * 10};
        reward = {(rescue level of critical point * 10) + (required time to rescue in critical point *
        70)};
        probability = {(reward - cost)/ reward};
        Add to listProbabilities ("unique identifier of rescue" ,"probability");
    sumAllPropability = SUM( listProbabilities);
    listAutomatonProbabilities = NewList();
    //define a new list for calculate new probabilities that uses by automaton;
    foreach unique identifier of rescue in listProbabilities
        Add to listAutomatonProbabilities ("unique identifier of rescue", "probability of
        rescue/sumAllPropability");
```

We removed the parameters which were not discussed in our model and improved the formulas. Getdistance () function is defined in order to calculate the distance between the real-time location of rescue team and the location of critical point.

The parameter of time taken to reach the desired location is achieved by dividing the distance by the average speed of each person for travel one meter which is equal to 5 m/s. So we divided the distance by 5 to calculate the average time to reach the critical point. The parameter of 'rescue level of critical point' was determined based on the level of rescue associated with the specifications of the critical point, and it has four values: high level, moderate level, low level, and bad level; they are shown with the numbers 0–4. Coefficients were obtained based on trial and error to refine the formula.

5. Now, the automaton first selects a team in a stochastic environment based on the obtained probabilities. The objective of the automata is to converge to the optimal operation. In other words, it converges toward selecting the best rescue team in the operational area to assign its identified relief task. So it is essential that each assistant agent of the search team be informed of the decisions of the other search teams working in the same operational area before delegating a task to a selected team of rescuers.

6. In real-world situations, it is important for each assistant agent to be informed of the decisions of the other neighbor agents. This can increase the performance of the rescue operation by avoiding the redundancy and conflict among different neighbor agents. In this regard, before allocating a specific task to a selected rescue team, each assistant agent should be informed of the decisions of its neighbor

search teams. Therefore, when an assistant agent wants to allocate a rescue operation to a rescue team, it should consider the importance of the selected team for neighbor search teams and if necessary, change its election. This is mainly because of the fact that the severely injured victims should be treated by the rescue teams sooner than other victims with fewer injuries. Perhaps this team can be a more vital selection for performing relief tasks of other critical places that by its neighbor searcher teams had evaluated.

7. The automaton evaluates the importance of the selected rescue team for itself (named spatial environment) and for the competing teams (named neighboring cells) based on the proposed learning model that is presented in 3.3. As a result, we have used a model for integrating neighborhood rules and factors of spatial environment for giving the reward or penalty to the selected action. Then, the probability vector may change based on the reward and penalty parameters as a linear vector with assistance of learning automata with L_{ReP} scheme. In fact, the purpose of each learning automaton designed for an assistant agent is to select a rescue team for allocating tasks, and this selection is not only the best choice for itself, but also the best choice for all search teams to allocate their tasks.
8. A key parameter in assigning spacial-temporal distributed tasks is time. Due to the importance of this parameter, we used a time period named *deadline* for managing the time on the performance of the automata. *Deadline* specifies the time duration in which an identified task can be assigned to the appropriate rescue team. Therefore, in the decision-making algorithm, first, *deadline* for each task is calculated based on its criticality. Then, with start of decision-making operation related to the task by the automaton, *deadline* variable also begins to calculate the time until it reaches its maximum calculated value. In other words, the defined *deadline* supervises the learning duration of the automata in making decision on task assignment. When the *deadline* expires, the following scenarios occur:
   - If the selected rescue team by the automaton was not chosen by any other neighbor search teams, then a task with priority 1 is assigned to the respective rescue team.
   - If the selected rescue team by the automaton was already chosen by a neighbor search team, then a task with priority 2 is assigned to the respective rescue team.
   - If the automaton fails to select an appropriate rescue team, due to any incidence (either interruption of communication or any unpredictable problem), this task is sent back to the IC by the agent. The IC makes an optimal decision for the corresponding task by using information of all teams and considering the situation of them on the map which are entered in central database.

   Each automaton assigned detected task and new detected tasks to the appropriate rescue team. Therefore, all agents have assigned their identified tasks to the rescue teams at the same time. In other words, all tasks are assigned simultaneously to all rescue teams.
9. Finally, the information related to the assigned tasks is sent to the intelligent assistant agent of the selected rescue teams and the rescue teams are informed of their tasks and the location of the assigned tasks and then set out to the assigned places according to their priority to rescue and take the victims.
10. The selected rescue teams receive their rescue tasks and move to the location of the assigned tasks. After assigning the rescue tasks to rescue teams, search teams move to next most critical points.

3.2 The proposed structure of CLA

As described earlier, for each intelligent assistant agent of search teams, one learning automaton is designed. A general schematic of using CLA in task allocation is shown in Fig. 2, and now Fig. 4 shows connections of the learning automata.

A VLSA with L_{ReP} scheme indicated by a quintuple $< \alpha, \beta, p, T(\alpha, \beta, p) >$, where '$\alpha$' is an action set with M actions (a set of rescue teams that identified by $R = \{R_1, R_2, \ldots, R_M\}$) and '$\beta$' is an final response set that is calculated by the proposed learning model, is applied in this article. In other words, the automaton evaluates the importance of the selected action for environment based on the proposed learning model that is presented in Sect. 3.3. We have two environments: spatial environment and neighboring cells, $\beta_{\mathrm{E-Spatial}}$ is spatial environment response and $\beta_{\mathrm{E-Neighbor}}$ is neighboring cells response. Finally, 'p' is the probability set with M probabilities, each being the probability of performing every action in the current internal automaton state.

3.3 Model for integration neighborhood rules and factors of spatial environmental

After a specific cell selects an action a_{i}, the behavior of the cell is evaluated according to the following scheme:

$$\beta = T_1 \odot T_2 \odot T_3 \tag{4}$$

where β is the response of the neighbors, T_1, T_2 and T_3 are three logical variables which are explained in Table 2, and

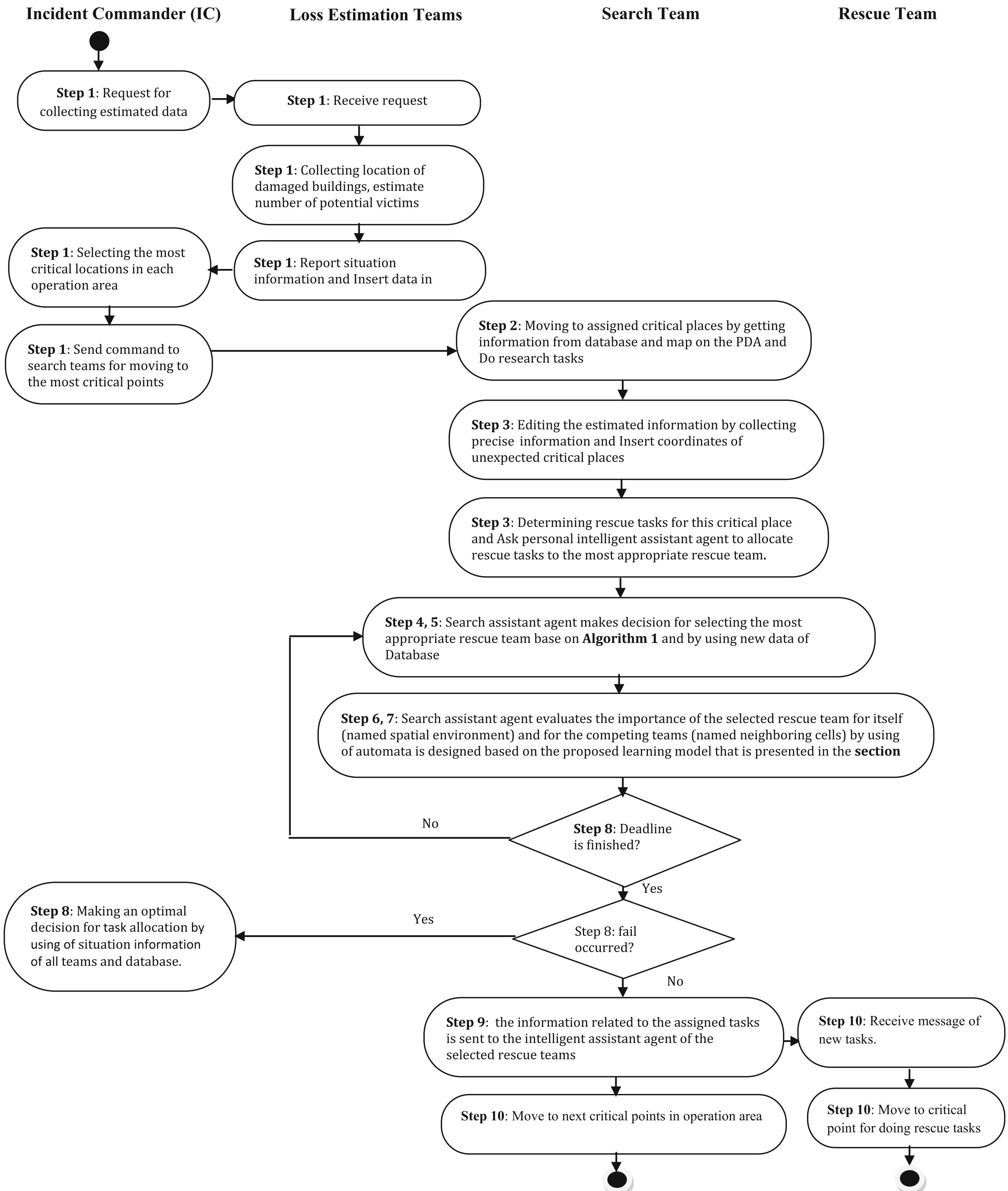


Fig. 3 Activity diagram for the steps of task allocations

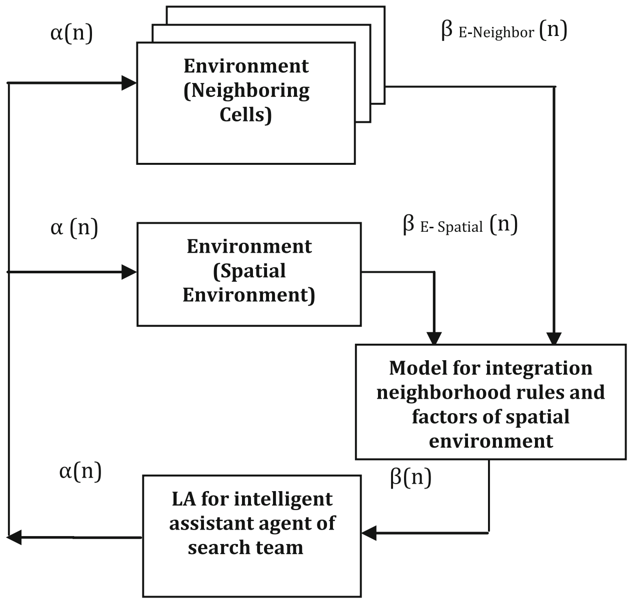


Fig. 4 The proposed structure of cellular learning automata for allocating spatial distributed tasks

$\odot$ represents the way that the logical variables are combined together to form the β.

Considering $\odot$, the β is determined as follows:

$$\beta = \begin{cases} 1 & T_1 = T_2 = T_3 = \text{true} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The value of the β can determine whether the response of the neighbors were desirable or not, where $\beta = 1$ corresponds to unfavorable response and $\beta = 0$ corresponds to favorable response.

In Table 2, the concept of vital selection needs to be defined. The automaton investigates its next optimal choice and its neighbors' next optimal choices using GIS functions and some important parameters such as distance. The automaton calculates "listProbabilities" variable parameter of algorithm (1) for itself and its neighbors for this purpose. If the obtained probability for the next selected optimal rescue team of any neighbors is less than the probability for its own next selected optimal rescue team, it is concluded that the current selected team may be a vital selection for the neighbors. It means that if a competitor neighbor wants to neglect the team, it may lose some elements such as time.

We applied the majority-minority rule for neighborhood effect (Schiff 2008). Table 3 shows the model for integration the neighborhood rules and the factors of spatial environment. In the proposed model, the factor of the spatial environment is defined based on the distance between the rescue team and critical point. An automaton gets reward from the environment if the selected rescue team is at the minimum distance from the critical point.

The probability coefficients of Table 3 have been obtained by performing several tests and based on the fact that the automaton does not converge quickly to the wrong action. The purpose of selecting these coefficients is evaluation of results of selected action by automata for spatial environment and neighboring cells.

Therefore, the automaton first selects a team in a stochastic environment based on the obtained probabilities. Then, it evaluates the importance of the selected team for itself (environmental factor) and the other competitor teams (computeβ). The automaton gets penalty from its rival neighbors if this option is vital selection for even one of the competing teams. Thus, with the probability of penalty equal to 0.5, it evaluates this selection as the vital selection for its rival neighbors and changes its optimal choice, and the second optimal choice would be its convergence criterion. In other words, the automaton learns to change its selection criteria in order to have better options in the next steps. However, with the probability of the reward equal to 0.5, it learns that its choice was the optimal choice (because it is likely that this selection may remain optimal choice for these automata). As a result, the learning automaton enables the agent to perform the task allocation by cooperating with other agents. Then, the probability vector is updated according to reward and penalty using a linear scheme L_{ReP} with $a = 0.4$ and $b = 0.1$. This process continues until the time duration defined in *deadline* is finished. Then, the task with priority 1 or 2 is assigned to the proper rescue team.

4 Implementation of geospatial simulation environment

4.1 Designing the test environment

To simulate the structure of STCP and to implement the spatial distributed intelligent agents, we have developed a geospatial simulation with C# programming language under the .NET framework (Crooks and Castle 2012; Fasli and Michalakopoulos 2006). ArcGIS Desktop software for designing desired GIS Shapfiles (Law and Collins 2013) and DotSpatial as a geographic visualization tool of .NET for working with GIS data (Parks 2011) are applied. Layers of blocks, parks and green spaces, rivers and the division of areas on the map also are applied. The scale of the map used in the simulator is $\frac{1}{5000}$. Information about the area, the profile of damaged buildings, specifications of rescue and search teams was entered in the database. To manage and access to data and information by agents, we implemented a SQL database and used its functions. Figure 5 shows the schematic diagram of the implemented simulation environment. As shown in the figure, the GIS functions are applied to add maps, layers and working with GIS data layers in the

Table 2 Description of logical variables for determining the response of the neighbors

	Condition
T_1	Is the selected rescue team the same as teams selected by my neighbors?
T_2	Is the selected rescue team located at the nearest distance from the competitor neighbors?
T_3	Is the selected rescue team a "*vital selection*" for the competitor neighbors?

Table 3 Model for integration neighborhood rules and factors of spatial environment

Neighborhood law	Environmental factors	Result for selects α_i action
Neighbor = 0	Reward	Reward
Neighbor = 0	Penalty	Reward = 0.15, penalty = 0.85
Neighbor = 1	Reward	Reward = 0.5, penalty = 0.5
Neighbor = 1	Penalty	Penalty

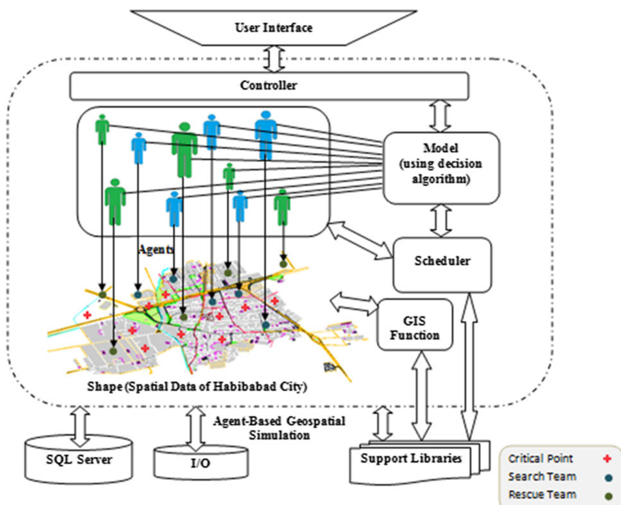


Fig. 5 The schematic diagram of the implemented simulation environment

Fig. 6 The designed geospatial simulation environment

simulator. The model element represents function related to create the object of the search and rescue teams with their behaviors, attributes and their respective functions on the map. The scheduler element represents the timing functions that its task is to manage the simultaneous implementation of search and rescue teams located on the map. The controller element also manages the activities of all the elements and shows other functions.

The area selected for this study was Habibabad city, one of Isfahan province cities located at 10-km northeast of Isfahan in Iran. The selected location is placed on the radius of the fracture. This is as one of the regions recently shaken by a 4.1-magnitude earthquake. The final output of the designed program for geographic simulation of spatially distributed intelligent agents is indicated in Fig. 6.

4.2 Automata performance evaluation

The learning process of four assistant agents for the search teams located in a similar operating area is shown in Fig. 7. As can be seen from the graphs, each agent has selected a rescue team for allocating its identified task according to the time limit of the critical point. In the diagram of Fig. 8, the learning process of the four assistant agents for the search teams located in a similar operating area is shown in the total time the scenario is executed. As can be seen from the diagram, agents related to search teams with the ID = 0, ID = 1, ID = 2 and ID = 3 selected an appropriate rescue team for the assignment of each of the identified relief tasks. Since the *deadline* monitors automata's performance, the time period of selecting the appropriate rescue team by the search team for assignment of the relief task with respect to the status of the task is different. As can be seen, for instance, the first relief task identified (T^3) by any of search teams was assigned to individual rescue teams. But for the assignment of the second identified relief task (T^2), the agent related to the search team with ID = 3 selected the

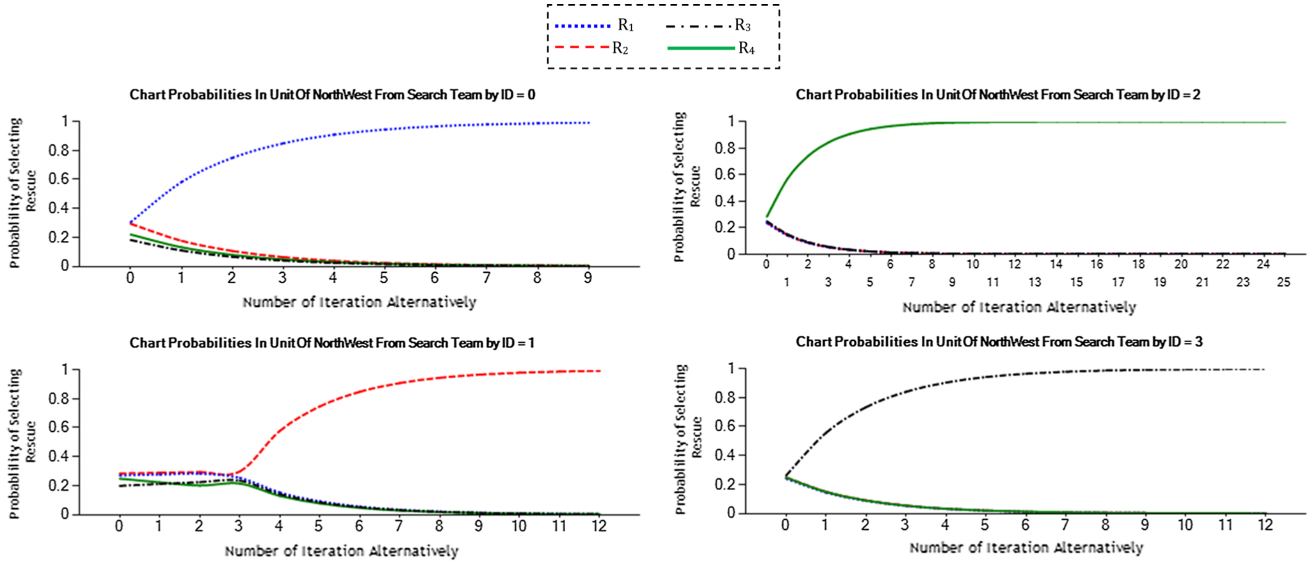


Fig. 7 The learning process of four assistant agents of the search teams located in a similar operating area for assigning one rescue task of operation area to the appropriate rescue team

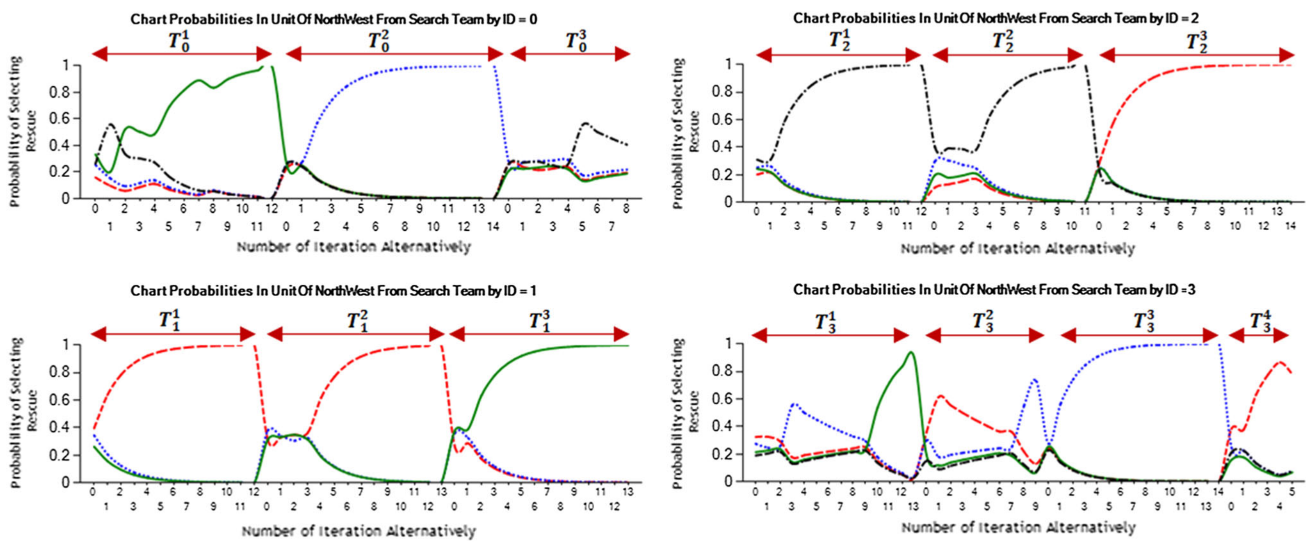


Fig. 8 The learning process of four assistant agents for the search teams located in a similar operating area for assigning all rescue tasks of operation area to the rescue teams

rescue team R_2. So, the agent gets penalty for this selection and reduces its probability because it was more critical to its neighbor (the search team with ID $=$ 1). According to time management by the *deadline* on the automata performance, the time expired for selecting the appropriate rescue team for assignment of the second identified task by the search team with ID $=$ 3. Therefore, the agent related to the search team with ID $=$ 3 has selected the rescue team R_1 (the team with the highest probability) for assigning its identified relief task and it assigned its task with priority 2 to the rescue team R_1 because the rescue team R_1 is selected with more probability by the search team ID $=$ 0.

5 Experimental results

In order to evaluate our proposed model in accordance with the auction and bidding model based on the contract net-work protocol (auction based) (Nourjou et al. 2011; Dias and Stentz 2000; Song et al. 2009; Hunsberger and Grosz 2000; Ham and Agha 2007; Smith 1980; Dias 2004; Dias et al. 2006; Dorigo 2005; Dorigo et al. 2013; Ham and Agha 2008; Gerkey and Mataric 2004; Liekna et al. 2012; Hussein and Khamis 2013) and with response threshold-based model (RTM) (Quiñonez et al. 2011a; Yasuda et al. 2014; Brutschy and Pini 2014; Cornejo and Dornhaus 2014; Fathy

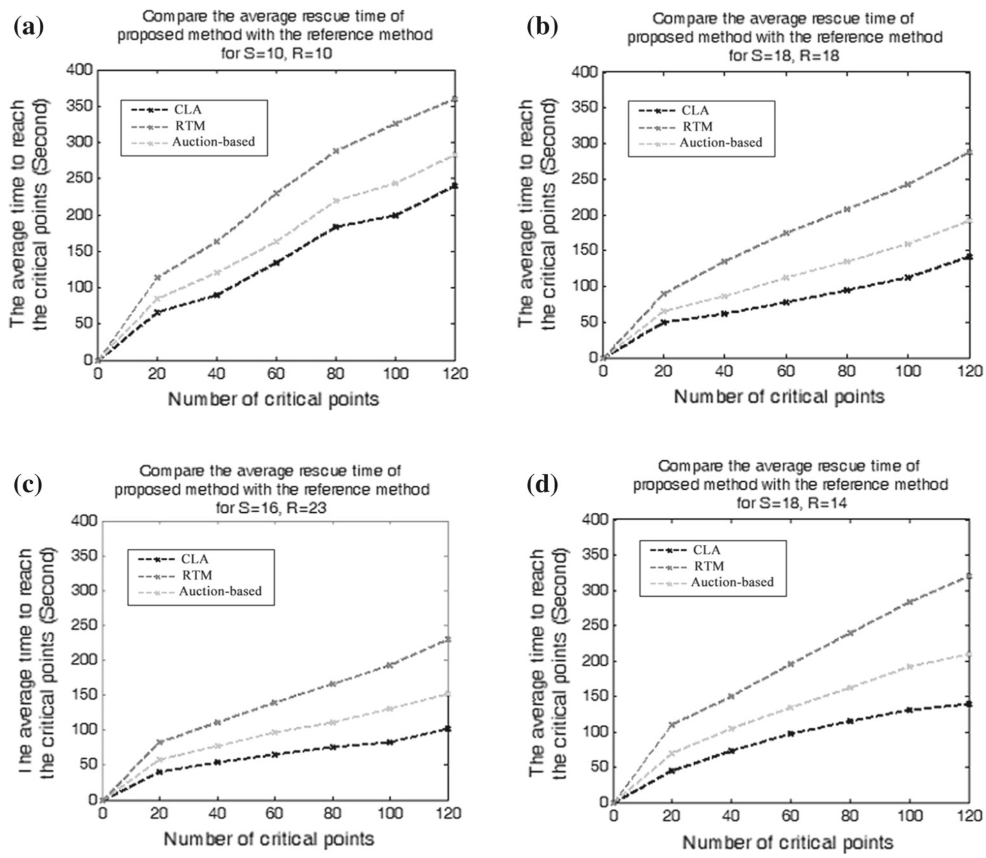


Fig. 9 Comparison of proposed model with the reference models: performance on the average rescue time in different states

Navid and Aghababa 2013; Labella et al. 2006; Campo and Dorigo 2007; Liu et al. 2007; Castelloh et al. 2013; Ferreira et al. 2008; Ikemoto et al. 2010) as the reference models, a number of different scenarios with different parameters were developed in the same geospatial simulation environment. In other words, we examined the algorithms output by the same input data and simulation environment. To this end, in the illustrated model in Fig. 4, the desired algorithm for modeling the agents and the results of running the algorithm are applied. For simulation and implementation of the algorithms, we first assumed that the number of critical points in each operational area and the estimated number of victims which were collected by the loss estimator teams were inserted in database and located on the map. After that, running the algorithm started with sending search teams to the more critical places in each area, and after completing the search operations, the search teams summoned the rescue teams (by presented task allocation algorithm) for performing the relief operation.

The input parameters used for different scenarios are as follows: the number of rescue (N_{Rescue}) and search (N_{Search}) teams, the number of critical places, and the learning rate. The learning rate is obtained by trial and error, and it is considered by $a = 0.4$ and $b = 0.1$. It should be noted that the outputs obtained from each scenario were the average of 20 times experiments. First, the main parameter the "the average rescue time" defined the average time it took a rescue team to reach a critical location. This parameter is achieved by dividing distance by the average speed of each person for travel one meter. Average speed of each person for travel one meter is equal to 5 m/s. We therefore divided the traveled distance by the rescue team to reach a critical point by 5 in order to calculate the reaching time to a critical location for a rescue team. Next, we calculated this time for all the critical locations. Finally, the average of this time for each scenario and for the implementation of each of the algorithms was computed. What is certain is that if the search teams can simultaneously select proper rescue teams to assign their tasks, the average rescue time will be less.

Figure 9 shows the evolution of this parameter for a set of critical points $\{20, 40, 60, 80, 100, 120\}$ by different scenario of search numbers and rescue numbers for CLA, auction-based, RTM algorithms.

(a) $L = \{N_{\text{Search}} = 10, N_{\text{Rescue}} = 10\}$
(b) $L = \{N_{\text{Search}} = 18, N_{\text{Rescue}} = 18\}$
(c) $L = \{N_{\text{Search}} = 16, N_{\text{Rescue}} = 23\}$
(d) $L = \{N_{\text{Search}} = 18, N_{\text{Rescue}} = 14\}$

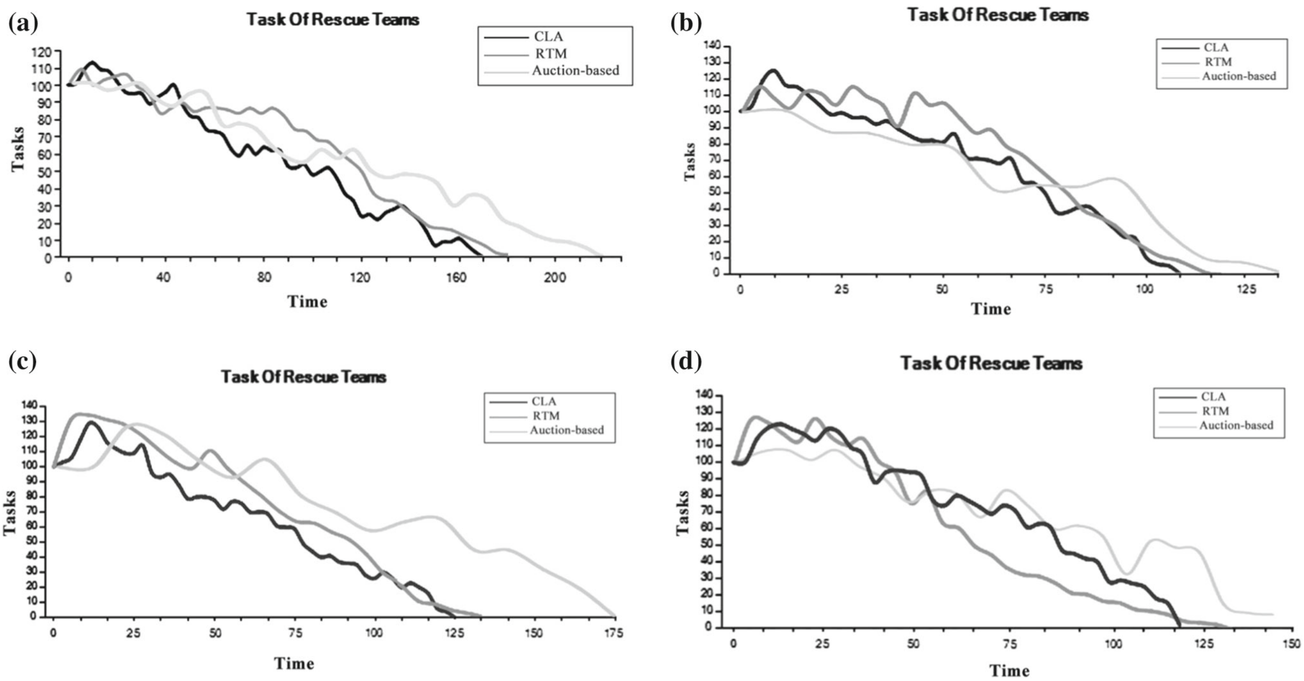


Fig. 10 Comparison of proposed model with the reference model: performance on the total run time in different states, with 100 critical points. **a** Compare the total run time of proposed model with reference methods for $S = 10$, $R = 10$. **b** Compare the total run time of proposed model with reference methods for $S = 18$, $R = 18$. **c** Compare the total run time of proposed model with reference methods for $S = 16$, $R = 23$. **d** Compare the total run time of proposed model with reference methods for $S = 18$, $R = 14$

The following graphs in Fig. 9 show the considerable progress in the average rescue time using the proposed model, which is the key parameter in improvement of performance management for the urban search and rescue (USAR) operations which contain locating, rescuing, and treating the injured people trapped in collapsed buildings.

Regarding the output of graphs, the RTM method has indicated the highest average time to reach a critical point in all four tested cases and CLA method the lowest. Also, by increasing the number of rescuers, the average time to reach the critical location has declined. By increasing the number of critical locations, the slope of RTM graph also increased; in other words, the average time to reach the critical location had more increased than in the other two methods, while in CLA method by the increase in the critical locations, the slope has incremented more mildly which shows CLA method successively performed the best task allocation among rescue teams. As a result, CLA method had the least amount of time to reach critical locations.

Figure 10 shows the evolution of the system performance on the total run time for task allocation of 100 critical points (new tasks are added in the database while new critical points are identified by search teams) among rescue teams by different scenarios of search numbers and rescue numbers. The result shows total running time of the proposed model for different scenarios has always been lower compared to the other two models. Also, total running time of proposed model is closer to auction and bidding method.

In order to do more accurate evaluation of our proposed model, a number of different scenarios with different number of search teams and rescue teams on 50 and 100 critical points were developed. The purpose of this evaluation was a closer study of the following three parameters: (1) average time to reach the critical points, (2) total displacement of rescue teams, and (3) the average efficiency of rescue teams. As shown in Table 4, the parameters of “average time to reach the critical points” and “total displacement of rescue teams” were improved. So all rescue teams could cover all critical areas with going through the minimum distance to make maximum use of time. The results indicate that we can have maximum utilization of the space and time in USAR operations by using the proposed model.

As shown in Table 4, the total distance traveled by the rescue teams in the proposed method compared to auction-based and RTM methods has declined substantially. This decreasing trend can be observed in each of the four tested scenarios. The decline shows that the search teams have always selected the appropriate rescue teams to assign their identified relief task. Therefore, the rescue teams have traveled a shorter distance to reach critical locations, resulting in less time waste. The average efficiency of rescue teams is another evaluated parameter which determines the average number of selected

Table 4 Scenarios for evaluating and comparing the performance among proposed allocation model and the reference models

Name of scenarios and approaches		Controls		Response for point = 50			Response for point = 100		
Scenario	Approach	The number of rescue teams	The number of search teams	Total displacement of rescue teams (m)	The average time to reach the critical points (s)	The average efficiency of rescue teams (%)	Total displacement of rescue teams (m)	The average time to reach the critical points (s)	The average efficiency of rescue teams (%)
Scenario1	CLA	10	10	8324.87	33.3	0.97	19,443.70	200.48	0.95
Scenario1	Auction-based	10	10	10,968.62	40.696	0.98	22,538.00	243.45	0.95
Scenario1	RTM	10	10	14,331.73	52.364	0.76	27,119.67	325.66	0.72
Scenario2	CLA	18	18	786.35	31.136	0.90	15,152.61	113.19	0.89
Scenario2	Auction-based	18	18	10,399.05	41.584	0.94	20,978.12	160.75	0.91
Scenario2	RTM	18	18	14,120.95	56.472	0.70	29,975.00	242.62	0.65
Scenario3	CLA	16	23	7852.73	30.428	0.93	13,004.24	78.23	0.90
Scenario3	Auction-based	16	23	11,095.86	44.388	0.98	22,777.34	122.57	0.96
Scenario3	RTM	16	23	16,557.44	66.224	0.77	30,103.09	215.80	0.74
Scenario4	CLA	18	14	8231.88	27.83	0.91	17,819.19	130.614	0.88
Scenario4	Auction-based	18	14	9778.83	39.116	0.97	24,145.37	192.80	0.95
Scenario4	RTM	18	14	14,942.54	59.768	0.52	29,616.32	283.40	0.60

rescue teams at any time for relief operation. By evaluating these parameters, we can assess the strength of each of the algorithms for the maximum usage of existing rescue teams. We are faced with a little reduction in efficiency associated with the parameter "the average efficiency of rescue teams" compared to auction-based model because in auction-based method, when an agent recognizes a task, it follows a protocol of messages in order to make a contact with other agents. The result of pursuing the protocol of messages is engaging all the agents in task allocation; however, the negative point of this act refers to the exceeding time to establish a contact that results in a significant increase in the average rescue time. This takes a great deal of time to get to a contact and heavily limits the application of auction based for allocating tasks among the agents. We hope to reduce this parameter in our proposed model in further experiments and with approving our model.

Result shows a bad performance of auction-based model and response threshold-based model compare to proposed model which is related to type of communication used in each model and both managing time and selecting best rescue team. Communication has important effect on the performance of coordination among different agents (Cayrpunar et al. 2008). The challenge in coordination among agents especially in critical environment is to discover "optimal" pieces of information to exchange without overload the communications bandwidth (Parker 2008). A mixture of explicit method (agents sense the effects of other agents' actions through their effects on the world.) and implicit methods (agents directly communicate with each other by some active means) demonstrates strengths in search and rescue experiments (Yan et al. 2013). A hybrid method in the proposed model for improving the performance is applied. Also, time constraint in search and rescue operation for task allocation is essential, so that selecting best path and selecting best rescue team to reach victims which should be found within some time is important. The learning automata task allocation algorithm guides the choice of future action by past responses; therefore, we use learning automata algorithm to select best rescue team and best path in the shortest time while both managing time and selecting best rescue team are not considered in the auction-based model and the response threshold-based model.

6 Discussion and conclusion

In this article, we have proposed a stochastic reinforcement-learning algorithm based on cellular learning automata and GIS data for solving the problem of coordination in multi-agent systems in assigning the spatial–temporal distributed tasks. In fact, by interaction between humans and intelligent agent equipped with the ability to learn, a method based on collaborative learning is presented for spatial–temporal allocation of tasks. The ultimate objective is to maximize the number of the rescued victims in the shortest time possible.

To evaluate the proposed algorithm, we designed the multi-agent geographical simulation and performed the number of different scenarios with the different number of rescue and search teams and some critical points in the simulated environment. As it is clear from the results, the proposed method makes maximum use of time and space in critical events. So all rescue teams could cover all critical areas by going through the minimum distance to make maximum use of time. As future work, combining new learning automata with other learning methods can be a strategy to continue this research. Moreover, making the simulation environment more realistic along with using geospatial information can be effective in obtaining better results.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Adams NM, Field M, Gelenbe E, Hand DJ et al (2008) The Aladdin project: intelligent agents for disaster management. In: IARP/EURON workshop on robotics for risky interventions and environmental surveillance

Akbari Torkestani J (2012) An adaptive learning automata-based ranking function discovery algorithm. J Intell Inf Syst 39:441–459

Ashish N, Eguchi R, Hegde R et al (2007) Situational awareness technologies for disaster response. In: Chen H et al (ed) Terrorism informatics: knowledge management and data mining for homeland security. Springer, Boston, MA, pp 517–544

Barzegar S, Davoudpour M, Meybodi MR et al (2011) Formalized learning automata with adaptive fuzzy coloured Petri net; an application specific to managing traffic signals. Sci Iran 18(3):554–565

Beigy H, Meybodi MR (2003) A self-organizing channel assignment algorithm: a cellular learning automata approach. Lecture notes in computer science. Springer, Berlin, pp 119–126

Beigy H, Meybodi MR (2004) A mathematical framework for cellular learning automata. Adv Complex Syst. doi:10.1142/S0219525904000202

Beigy H, Meybodi MR (2008) Asynchronous cellular learning automata. Automatica 44(5):1350–1357

Brutschy A, Pini G (2014) Self-organized task allocation to sequentially interdependent tasks in swarm robotics. Auton Agents Multi Agent Syst 28(1):101–125

Campo A, Dorigo M (2007) Efficient multi-foraging in swarm robotics. In: Advances in artificial life, lecture notes in computer science, vol 4648, pp 696–705

Castelloh E, Yamamoto T, Nakamura Y, Ishiguro H (2013) Task allocation for a robotic swarm based on an adaptive response threshold model. In: 13th International conference on control, automation and systems. Kimdaejung Convention Center, Gwangju, Korea, pp 259–266

Cayrpunar O, Tavl B, Gazi V (2008) Dynamic robot networks for search and rescue operations. In: International workshop on robotics for risky interventions and surveillance of the environment

Cornejo A, Dornhaus A et al (2014) Task allocation in ant colonies. Distrib Comput 8784:46–60

Crooks AT, Castle CJE (2012) The integration of agent-based modelling and geographical information for geospatial simulation. In: Agent-based models of geographical systems. Springer, Dordrecht, pp 219–251

Dias B (2004) Traderbots: a new paradigm for robust and efficient multirobot coordination in dynamics environments. Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh

Dias B, Stentz A (2000) A free market architecture for distributed control of a multirobot system. In: 6th International conference on intelligent autonomous systems, pp 115–122

Dias MB, Zlot R, Kalra N, Stentz A (2006) Market-based multirobot coordination: a survey and analysis. Proc IEEE 94:1257–1270

Dorigo M (2005) SWARM-BOT: an experiment in swarm robotics. In: Arabshahi P, Martinoli A (eds) 2005 IEEE swarm intelligence symposium, pp 192–200

Dorigo M, Floreano D, Gambardella LM, Mondada F, Nolfi S, Baaboura T et al (2013) A novel concept for the study of heterogeneous robotic swarms. IEEE Robot Autom Mag 20(4):60–71

Dos Santos F, Bazzan ALC (2011) Towards efficient multiagent task allocation in the robocup rescue a biologically-inspired approach. Auton Agents Multi Agent Syst 22:465–486

Esnaashari M, Meybodi MR (2008) A cellular learning automata based clustering algorithm for wireless sensor networks. Sens Lett 6:723–735

Esnaashari M, Meybodi MR (2010) Dynamic point coverage problem in wireless sensor networks: a cellular learning automata approach. Ad Hoc Sens Wirel Netw 10:193–234

Fasli M, Michalakopoulos M (2006) Developing software agents using.NET. University of Essex Department of Computer Science, Colchester

Fathy Navid AH, Aghababa AB (2013) Cellular learning automata and its applications. In: Computer and information science, numerical analysis and scientific computing. Publish with INTECH. doi:10.5772/52953

Fayyoumi E, Oommen BJ (2009) Achieving micro aggregation for secure statistical databases using fixed structure partitioning based learning automata. IEEE Trans Syst Man Cybern B Cybern 39(5):1192–1205

Ferreira PR, Boffo FS, Bazzan ALC (2008) Using swarm-GAP for distributed task allocation in complex scenarios. Massively multiagent technology, LNCS, vol 5043, pp 107–121

Ferreira PR Jr, Dos Santos F, Bazzan ALC, Epstein D, Waskow SJ (2009) RoboCup rescue as multi agent task allocation among teams: experiments with task interdependencies. Auton Agents Multi Agent Syst 20(3):421–443

Gerkey BP, Mataric MJ (2003) Multi-robot task allocation: analyzing the complexity and optimality of key architectures. In: IEEE international conference on robotics and automation. doi:10.1109/ROBOT.2003.1242189

Gerkey BP, Mataric MJ (2004) A formal analysis and taxonomy of task allocation in multi-robot systems. Int J Robot Res 23(9):939–954

Godoy J, Gini M (2013) Task allocation for spatially and temporally distributed tasks. In: Proceedings of the 12th international conference IAS-12, AISC 194. Springer, Berlin, Heidelberg, pp 603–612

Grinshpoun T, Grubshtein A et al (2013) Asymmetric distributed constraint optimization problems. J Artif Intell Res 47:613–647

Ham M, Agha G (2007) Market-based coordination strategies for large-scale multi-agent systems. Syst Inf Sci Notes 2(1):126–131

Ham M, Agha G (2008) A study of coordinated dynamic market-based task assignment in massively multi-agent systems. In: Massively

multiagent technology, lecture notes in computer science, vol 5043, pp 43–63
Hunsberger L, Grosz BJ (2000) A combinatorial auction for collaborative planning. In: Proceedings of the fourth international conference on multiagent systems, ICMAS, pp 151–158
Hussein A, Khamis A (2013) Market-based approach to multi-robot task allocation. In: International conference on individual and collective behaviors in robotics, pp 69–74
Ikemoto Y, Miura T, Asama H (2010) Adaptive division-of-labor control algorithm for multi-robot systems. J Robot Mechatron 22(4):514–525
Jain S, McLean CR (2004) An architecture for modeling and simulation of emergency response. In: Proceedings of the 2004 IIE conference
Jinguo L, Yuechao W, Bin L, Shugen M (2007) Current research, key performances and future development of search and rescue robots. Front Mech Eng China 2:404–416
Kalra N, Martinoli A (2006) Comparative study of market-based and threshold-based task allocation. Distrib Auton Robot Syst 7:91–101
Khalil KM, Abdel-Aziz MH, Nazmy MT, Salem AM (2008) The role of artificial intelligence technologies in crisis response. In: Mendel conference, 14th international conference on soft computing
Khalil KM, Abdel-Aziz MH, Nazmy MT, Salem AM (2009) Multi-agent crisis response systems—design requirements and analysis of current systems. In: Fourth international conference on intelligent computing and information systems. Egypt, Cairo, pp 920–925
Koes M, Nourbakhsh I, Sycara K (2005) Heterogeneo multirobot coordination with spatial and temporal constraints. In: Proceedings of the twentieth national conference on artificial intelligence (AAAI), pp 1292–1297
Labella TH, Dorigo M, Deneubourg JL (2006) Division of labor in a group of robots inspired by ants' foraging behavior. ACM Trans Auton Adapt Syst 1(1):4–25
Lakshmivarahan S (1981) Learning algorithms: theory and applications. Springer, Berlin
Law M, Collins A (2013) Getting to know ArcGIS for desktop. Environmental Systems Research Institute, Redlands, pp 19–30
Leite AR, Enembreck F, Barthès JA (2014) Distributed constraint optimization problems: review and perspectives. Expert Syst Appl 41:5139–5157
Liekna A, Lavendelis E, Grabovskis A (2012) Experimental analysis of contract net protocol in multi-robot task allocation. Appl Comput Syst 13(1):6–14
Liu W, Winfield A, Sa J, Chen J, Dou L (2007) Towards energy optimization: emergent task allocation in a swarm of foraging robots. Adapt Behav 15(3):289–305
Maheswaran RT, Rogers CM, Sanchez R, Szekely P (2010) Human-agent collaborative optimization of real-time distributed dynamic multi-agent coordination. In: 9th International conference on autonomous agents and multi-agent systems. Canada, Toronto, pp 49–56
Massaguer D, Balasubramanian V, Mehrotra S, Venkatasubramanian N (2006) MultiAgent simulation of disaster response. In: ATDM workshop in AAMAS, Japan
Meybodi MR, Kharazmi MR (2004) Application of cellular learning automata to image processing. J Amirkabir 14(56):1101–1126
Misra S, Oommen BJ (2009) Using pursuit automata for estimating stable shortest paths in stochastic network environments. Int J Commun Syst 22(4):441–468
Misra S, Abraham KI, Obaidat MS, Krishna PV (2009) LAID: a learning automata-based scheme for intrusion detection in wireless sensor networks. Secur Commun Netw 2(2):105–115
Moghiss V, Meybodi MR, Esnaashari M (2010) An intelligent protocol to channel assignment in wireless sensor networks: learning automata approach. In: International conference on information networking and automation, pp V1-338–V1-343
Murakami Y, Minami K, Kawasoe T, Ishida T (2002) Multi-agent simulation for crisis management. In: IEEE workshop on knowledge media networking
Najim K, Poznyak AS (1994) Learning automata: theory and applications. Pergamon, Oxford
Narendra KS, Thathachar MAL (1989) Learning automata: an introduction. Prentice-Hall, Englewood Cliffs
Neumann JV (1996) The theory of self-reproducing automata. In: Burks AW (ed) The theory of self-reproducing automata. University of Illinois Press, Urbana, London
Nourjou R, Hatayama M (2011) Intelligent GIS for spatial cooperation of earthquake emergency response. Ann Disaster Prev Res Inst Kyoto Univ 54(B):29–34
Nourjou R, Hatayama M, Tatano H (2011) Introduction to spatially distributed intelligent assistant agents for coordination of human-agent teams' actions. In: Proceedings of the 2011 IEEE international symposium on safety, security and rescue robotics, Kyoto, Japan, pp 251–258
Nourjou R, Smith SF, Hatayama M, Szekely P (2014a) Intelligent algorithm for assignment of agents to human strategy in centralized multi-agent coordination. J Softw 9(10):2586–2597
Nourjou R, Szekely P, Hatayama M, Ghafory-Ashtiany M, Smith SF (2014b) Data model of the strategic action planning and scheduling problem in a disaster response team. J Disaster Res 9(3):381–399
Nourjou R, Hatayama M, Smith SF, Sadeghi A, Szekely P (2014c) Design of a GIS-based assistant software agent for the incident commander to coordinate emergency response operations. In Workshop on robots and sensors integration in future rescue information system
Parker LE (2008) Multiple mobile robot systems. In: Springer handbook of robotics. Springer, pp 921–941
Parks A (2011) Using the open source DotSpatial GIS library to create tasks for the data for environmental modeling (D4EM) system. Research Triangle Institute International, North Carolina
Quiñonez Y, De lope J, Maravall D (2011a) Bio-inspired decentralized self-coordination algorithms. In: 4th International work-conference on the interplay between natural and artificial computation, IWINAC, La Palma, Canary Islands, Spain. Springer, Berlin, Heidelberg, pp 30–39
Quiñonez Y, Maravall D, De lope J (2011b) Stochastic learning automata for self-coordination in heterogeneous multi-tasks selection in multi-robot systems. In: 10th Mexican international conference on artificial intelligence, MICAI 2011, Puebla, Mexico. Springer, Berlin, Heidelberg, pp 443–453
Rahwan T, Jennings NR (2007) An algorithm for distributing coalitional value calculations among cooperating agents. Artif Intell J 171:535–567
Rasekh A, Vafaeinezhad AR (2012) Developing a GIS based decision support system for resource allocation in earthquake search and rescue operation. In: ICCSA 2012, part II, LNCS 7334. Springer, Berlin, Heidelberg, pp 275–285
Russell S, Norvig P (1995) Artficial intelligence: a modern approach. Prentice-Hall, Saddle River
Scerri P, Farinelli A, Okamoto S, Tambe M (2005) Allocating tasks in extreme teams. In: Proceedings of the fourth international joint conference on autonomous agents and multiagent systems, pp 727–734
Schiff JL (2008) Cellular learning automata: a discrete view of the world. Wiley, Chichester
Schoenharl T, Madey G et al (2006) WIPER: a multi-agent system for emergency response. In: Proceedings of the 3rd international ISCRAM conference, Newark, NJ, USA
Schurr N, Marecki J, Lewis JP, Tambe M, Scerri P (2009) The DEFACTO system: coordinating human-agent teams for the future

of disaster response. Published Articles of CREATE Research Archive
Shiroma P, Campos M (2009) CoMutaR: a framework for multi-robot coordination and task allocation. In: IEEE/RSJ international conference on intelligent robots and systems, pp 4817–4824
Smith RG (1980) The contract net protocol: high-level communication and control in a distributed problem solver. IEEE Trans Comput C–29(12):1104–1113
Song T, Yan X, Liang A, Chen K, Guan H (2009) A distributed bidirectional auction algorithm for multirobot coordination. In: IEEE international conference on research challenges in computer science, pp 145–148
Suarez Baron SA (2010) Dynamic task allocation and coordination in cooperative multi-agent environment. Doctoral Thesis of Programming in Technology, Dissertation, University of Girona
Talabeigi M, Forsati R, Meybodi MR (2010) A hybrid web recommender system based on cellular learning automata. In: IEEE international conference on granular computing, San Jose, California, pp 453–458
Testlin ML (1961) On the behavior of finite automata in randim media. Autom Remote Control 22(10):1210–1219
Theraulaz G, Bonabeau E, Deneubourg J (1998) Response threshold reinforcement and division of labour in insect societies. R Soc Lond Ser B Biol Sci 265:327–332
Vafaeinezhad AR, Alesheikh AA, Hamrah M, Nourjou R, Shad R (2009) Using GIS to develop an efficient spatio-temporal task allocation algorithm to human groups in an entirely dynamic environment case study: earthquake rescue teams. In: Computational science and its applications—lecture notes in computer science, vol 5592, pp 66–78
Yan Z, Jouandeau N, Cherif AA (2013) A survey and analysis of multi-robot coordination. Int J Adv Robot Syst 10:399
Yang Y, Zhou C, Tian Y (2009) Swarm robots task allocation based on response threshold model. In: IEEE international conference on autonomous robots and agents, pp 171–176
Yasuda T, Kage K, Ohkura K (2014) Response threshold-based task allocation in a reinforcement learning robotic swarm. In: IEEE 7th International workshop on computational intelligence and applications, Hiroshima, Japan, pp 189–194