IJICTR International Journal of Information & Communication Technology Research

ITRC

Volume 7- Number 1- Winter 2015 (29-39)

A Model Based on Cellular Learning Automata for Improving the Intelligent Assistant Agents & Its Application in Earthquake Crisis Management

Maryam Khani
Department of Mechatronic
Islamic Azad University, South Tehran Branch
Tehran, Iran
st_m_khani@azad.ac.ir

Ali Ahmadi
Department of Electrical and Computer
K.N. Toosi University of Technology
Tehran, Iran
ahmadi@kntu.ac.ir

Maryam Khademi
Department of Applied Mathematics
Islamic Azad University, South Tehran Branch
Tehran, Iran
khademi@azad.ac.ir

Received: June 10, 2014-Accepted: January 7, 2015

Abstract— **Spatial-temporal coordination problem (STCP) plays a critical role in urban search and rescue (USAR) operations. Artificial Intelligence has tried to tackle this problem by taking advantage of multi-agent systems, GIS, and intelligent algorithms to enhance the task allocation by establishing collaboration between human agents and intelligent assistant agents. This paper presents a model based on cellular learning automata (CLA) to improve the teamwork interaction between human-agent teams in performing the distributed tasks. In this model, the main objective is to add the learning ability to the assistant agents in a way that they can guide human-agent toward the optimal decision(s). The effectiveness of the proposed model is evaluated on different scenarios of an earthquake simulation. Results indicate that the proposed model can significantly improve the rescue time and the maximum distance traveled by the rescue teams.**

Keywords- Spatial-Temporal Coordination, Human-agent Interaction, Multi Agent System, Cellular Learning Automata, Earthquake Emergency Response, GIS.

I. INTRODUCTION

In the recent years, we have witnessed a growing number of natural disasters which have threatened the human safety. Earthquake is a typical paradigm of the destructive side of nature which still kills so many people all around the world. Therefore, USAR is of a great importance in saving peoples' lives. USAR contains locating, rescuing and treating the injured people trapped in collapsed buildings. In this operation, a collection of software agents, robots, rescue teams, crisis managers and crisis management organizations are interacting with each other to provide necessary assistance in a short period of time. Therefore, the main task of USAR operations is to determine who should do what, when and where.

Lots of efforts have been expended to enhance the performance of crisis management process [1, 2, 3, 4, 5, 6]. Crisis management involves multiple organizations and teams, geographically distributed operations, and its domain is characterized by: huge amount of data, uncertainty, ambiguity, multiple stakeholders with different aims and objectives, limited resources, and a necessary requirement for distributed control and decision making [4]. Crisis management systems design should include: 1) filtering and data fusion methods, 2) decision-making and machine learning methods, 3) manage the interaction between multiple actors by design of interaction mechanism methods such as multi-agent systems, 4) large studies of system architecture and information exchange topologies [1, 4].

One of crisis management process is the use of multi-agent systems. A multi-agent system (MAS) is a system composed of multiple interacting intelligent agents within an environment. The agents in a multi-agent system could equally well be robots, humans, and may contain combined human-agent teams. Multi-agent systems can be used to solve problems which are difficult or impossible for an individual agent to solve such as crisis response. In MAS, agent characteristics are autonomy, local view of environment, and capability of learning, planning, coordination and decentralized decision making. Another important research field in crisis response is the agent-based modeling and simulation. Several multi-agent systems have been introduced over the past decade for managing the hazardous events and simulating emergency responses. Some of these multi-agent systems include [1, 2, 7, 8, 9, 10, 11]: DrillSim, ALADDIN, RoboCup Rescue, FireGrid, Wiper, and DEFACTO.

Many studies have been made in the past to examine the interaction of human with a team of software agents [12, 13, 14, 15]. For example, Fong et al. [16] introduced an approach for remote control of multiple robots with the assistance of human-robot interaction. The efforts relevant to coordination of human-robot teams in space exploration, humans-robots-agents teamwork collaboration for teaching relief tasks in the incidence of a crisis have also been taken into account [14, 17, 18]. Previous works reported significant progress on the improvement of human-agents teamwork through integrated architectures designed based on proxy [14], adjustable autonomy agent (AA) [17], and human-agent dialogue [12].

Despite the fact that mentioned studies have made a significant contribution to the field, they still suffer from two major limitations. First, the current studies in remote human-agents interaction do not directly utilize the human perception for solving specific problems yet. Although several techniques exist that provide remote visualization through video streams, these methods do not utilize human intuition [16]. Second, the agent team is not flexible enough. On one hand, there are a number of cases that human users should make most of decisions. On the other hand, the role of human agents on decision-making may be completely eliminated in some cases. To improve the above disadvantages, in [19], the authors combined software proxy architecture and a three-dimensional visualization system in DEFACTO.

In general, the main motivation of researchers in this domain is to provide decision-making support systems to facilitate the coordination of the people in real-world situations. Such systems require synchronization, monitoring, planning, scheduling, management of uncertain data and distributiveness [20]. Therefore, researchers have tried to develop fundamental multi-agent systems from decision-making support systems to improve the performance of human activities in the environment. It is worth noticing that this method also suffers from two problems. First, the process of decision-making by the agents is not flexible. Second, there is no interaction among the population of agents. Task allocation is still an active research area [18] and several numbers of recent publications dedicated to this topic [21, 22, 23, 24, 25].

To address the above issues, this paper aims to apply a stochastic algorithm based on cellular learning automata to design the architecture of distributed intelligent agents in a way that the agents can make the best decision for task allocation.

In order to better describe our model, this paper is organized as follows: description of the proposed model is presented in section II. In section III, we propose our cellular learning automata based model. Section IV describes the implementation of a geospatial simulation environment for spatially distributed intelligent assistant agents using Visual Studio, software on .Net framework and C# programming language. Simulation results and discussions are given in section V.

II. Description of the Proposed Model

A. Modeling the problem

The very first step in proposing an effective solution to the spatial-temporal coordination problem is to enhance the modeling phase. In the following, we explain the task, dependent operations, required specifications, and working procedure of our model.

At the beginning of the process, the information related to the location of critical places and predicted number of victims is collected by the loss estimator teams. Afterwards, the search teams are deployed to the most critical locations of the environment by receiving order from the command center. In other words, at the beginning of the process command center selects the most critical locations based on collected information and sends messages to the search teams.The search teams are personally placed at the critical places and more information like: actual amount of damages, actual number of victims, level of relief needed, time limit to require relief, dangers of catching fires, collapse debris, flooded, etc. are collected by them. They also edit the estimated information and add the coordinates of unexpected critical places as new data in the system. The search team members also use their personal experience to estimate the level of relief needed for each place. Then, they enter all updated information in the central database with the help of intelligent assistant agent. Thus, this information will be shared with all teams involved in the critical incident and will help them to make the best decisions. The assistant agents of all search teams could be informed of the precise information relevant to the critical places through connection to 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 his appropriate rescue team according to the new entered data as well as important parameters [26, 27].

According to the explained process, existing relationships between activities defined. “Enable” relationship and “Equality” relationship that exist among activities presented in the simplified structure of the STCP in reference [26]. Enable" relationship makes possibility of performing for an action after completing another action; and “Equality” relationship which means certain actions are not related to a specific team, can be performed by other teams too [26]. In addition to the above relationships, we need to define some extra relationships like “Dependency” and “Collaboration”. Because, before allocating the identified relief task to the selected rescue team, it is essential that this agent be informed of the decisions of the other search teams located in the same area. So the assistant agent must consider the importance of the selected rescue team for other critical points and those severely injured in the calculations because more severe injured people must be rescued sooner. Thus, the selected rescue team may be a more critical choice for another critical point detected by another search team.

As a result, “Dependency” relationship shows the dependency between the new tasks identified by the search team, and “Collaboration” relationship shows the necessity of the relationship between the search team members for interaction development. Figure 1 shows an improved structure for STCP:

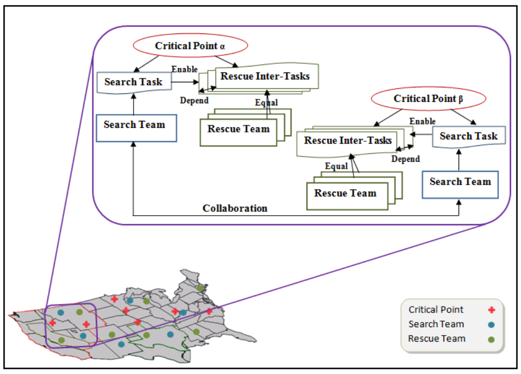

Fig.1. Modeling of Spatial - Temporal Coordination Problem from USAR operations for two spatial points α and β.

B. *Properties of the proposed model*

Generally speaking, the crisis management systems require prompt receiving, filtering and summarizing of the information, planning, scheduling, assigning tasks, collaborating, sharing information and making appropriate decisions in real time. Therefore, in order to solve the STCP problem, designing a crisis response system should cover crisis response domain requirements.

In the following, we mention these requirements. Some of these requirements have been derived from the other publications and some others have obtained from the study conducted in the present paper. These requirements are listed below:

(a) The crisis management system could answer the main question of the research, which is as follows: How can we achieve a more effective way to improve the coordination between humanitarian teams in order to allocate the relevant tasks in a dynamic environment with spatial characteristics.

(b) The interaction between human and the individual assistant agent in order to perform the fundamental operation: it supports information acquisition, information analysis, decision making and action selection.

(c) Distributed Control [1].

(d) Having a mixed-initiative planning by which the human can actually make strategy decision in high level and the agent also makes tactical decisions based on those [26].

(e) There are some risks for communication [1].

(f) Collecting uncertain and sporadic information which are required to decision making processes [1].

(g) Managing uncertainty information about results of search and rescue actions in operation area.

(h) The need for geographic data management and data sharing.

(i) The system components are designed so that they can adapt themselves to the environment changes [1].

(j) The ability to learn from experience [1].

(k) Flexibility [1].

(l) Teams act in domains whose dynamics may cause new tasks to appear and existing ones to disappear.

(m) Perform multiple tasks subject to their resource limitation.

(n) Definition of common functions to perform each task, but with differing levels of capability.

(o) Perform some tasks in these domains are inter-task (must be simultaneously executed).

The specifications of the proposed multi-agent system that cover all crisis response domain requirements for solving STCP are classified as follows:

1- Multi-agent system specifications:

Multi-agent system architecture, communication and collaborative between agents, agent diversity (our assumption in this article is to focus on only two types of teams: search and rescue teams).

2- Cooperation human and intelligent agent:

Using intelligent assistant agents, interaction between human and intelligent assistant agents, intelligent assistant agent behavior modeling.

3- Data management:

Information management (through a central database and local database), supporting GIS data.

4- Coordination:

Task allocation, cooperation to improve the rescue time; reduce deaths and injuries, collaborative decision making.

5- Learning

Learning of agent to perform tasks, agents monitoring.

6- Others specification

Adapting with methods that is based on extreme teams [22], flexible planning.

III. Proposed Cellular Learning Automata Based Model

A. *Cellular learning automata*

Learning Automata (LA) is a sophisticated reinforcement-learning model for decision making in stochastic and unknown environments [28]. LA is capable of learning the optimal action, among a set of finite actions, by repeating a two-step process: (i) at each time step, 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. This interaction between LA and the environment can guide the LA toward selecting the optimal action. LAs have been successfully used in many applications such as intrusion detection in sensor networks [32], database systems [33], solving shortest path problem in stochastic networks [34], channel assignment in wireless sensor networks [35], managing traffic signals [36], and ranking function discovery algorithm [37].

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 [29]. 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 like 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 $T: (s_i^t, \phi) \rightarrow \Delta$ 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.

A Cellular Learning Automata is a CA in which each cell contains one or more learning automata. The state of every cell is determined on the basis of action probability vector of the learning automata (or group of learning automata) residing in that cell. The initial value of this state may be chosen based on the past experience or at random. In the second step, the rule of CLA determines the reinforcement signal to each learning automaton. In other words, each cell is then evolved based on its experience and the behavior of its neighborhood cells. For each cell, the neighboring cells constitute the environment of the cell. Finally, each learning automaton updates its action probability vector on the basis of supplied reinforcement signal and the chosen action. This process continues until the desired result is obtained [29]. The CLAs have been used in many applications [29] such as image processing [38], channel assignment in cellular networks [39], call admission control in cellular networks[40] and sensor networks [41], dynamic point coverage problem in wireless sensor networks[42], and hybrid web recommender system[43].

In this paper, the variable structure learning automata with L_{ReP} (Linear Reward-e-Penalty) scheme are used for each cell. According to the variable structure learning automata, we use the following relations to calculate the probability vector for the desirable and undesirable responses from the environment:

1) The environment favorable response (β=0):

$$P_i(n+1)=P_i(n)+a(1-P_i(n)) \qquad (1)$$

$$P_j(n+1)=(1-a)P_j(n) \qquad \forall j \;\; j\neq i$$

2) The environment unfavorable response (β=1):

$$P_i(n+1)=(1-b)P_i(n) \qquad (2)$$

$$P_j(n+1)=(b/(r-1))+(1-b)P_j(n) \qquad \forall j \;\; j\neq i$$

In formula (1) and (2), 'a' is reward parameter, 'b' is penalty parameter, and 'r' is number of possible actions. When a=b, the automaton is called L_{RP}. If b=0 the automaton is called L_{RI} and if 0<b<<a<1, automaton is called L_{ReP}.

B. *A stochastic reinforcement learning model based on CLA*

The spatial-temporal coordination problem can be represented by a set of K tasks that denotes by L={l_1, l_2, ..., l_K} with temporal and spatial constraints, they have to be dynamically identified by a set of N search teams that shows by S={S_1, S_2, ..., S_N} and delivered to a set of M rescue teams that shows by R={R_1, R_2, ..., R_M} in the same operational areas.

1) Different stages of distributed “Spatial-temporal” task allocation

When search teams are personally placed at the critical places, more accurate information is collected and estimated information is edited by them. Subsequently, the coordinates of unexpected critical places as new data in the system are added and after selecting appropriate rescue team, assigned them to a selected rescue team as interdependent tasks. So we have implemented the new Dependence relationship between the tasks in our model.

1. For each assistant agent in the search team, a variable structure learning automaton (VLSA) with L_{ReP} scheme was utilized. The Linear Reward-e-Penalty scheme is a simple linear model, but it is an optimal model. Degree of freedom L_{ReP} is greater than the linear model L_{R-I}, because L_{ReP} has two parameter a, b. In addition, this model is simpler than nonlinear models. So by using this method we were able to quickly achieve the desired convergence.

Each automaton is responsible for assigning an appropriate rescue team to a certain task. At the beginning of the operation, the automaton determines the selection probability of each rescue team based on the information of the existing rescue teams in the operational area as well as new information obtained from the critical points. The automata consider different parameters like the distance between rescue team and critical point, time required for relief, rescue level of critical point (shows level of rescue at critical point that depends on the amount of damage at that point, the numbers of people are injured, the level of injuries to peoples, the possibility of a fire, power cuts, etc), critical point status, the rescue team's abilities (Including all ability of rescue team for relief: number of rescue team, the equipment, team energy with respect to the number of tasks that have done, the skill level of the team), the number of unfinished tasks assigned to rescue teams, and etc. to calculate the initial selection probability of each rescue team. Therefore, the initial selection probabilities of different rescue teams are not equal. Algorithm (1) used for this purpose.

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 towards 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.

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

when search operation in critical point **finished**
rescues= select rescues in operation area;
listProbabilities = define a new list for calculate probabilities of select rescue teams;
foreach rescue **in** rescues
distance = distance between real-time location of rescue team and location of critical point;
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 “unique identifier of rescue” and “probability” to listProbabilities;
sumAllPropability = calculate the sum of all probabilities of the listProbabilities;
listAutomataProbabilities = define a new list for calculate new probabilities that uses by automata (that was utilized in assistant agent);
foreach unique identifier of rescue **in** listProbabilities
add “unique identifier of rescue” and “probability of rescue/sumAllPropability” to listAutomataProbability;

2. When an earthquake occurs, a crisis region is divided into several areas, each of which is called an operational area. A number of search teams and rescue teams are deployed in each area of operations. In the proposed model, search teams in each operational area are defined as neighbors. In other words, the search teams are identified as the cells of the CA based on their operational regions. In order to coordinate the operation of the agents in the environment, the concept of cellular learning automata is used in the proposed model to share the information among the agents that are working in the same operational area of the environment. So, the new collaboration relationship is implemented among search teams. 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 between 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, changed its election. This is

mainly because of the fact that the severely injured and wounded victims should be treated by the rescue teams sooner than other victims with fewer injuries. Perhaps this team can be a more critical selection for performing relief tasks of other critical places that by its neighbor searcher teams had evaluated.

3. The automaton evaluates the importance of the selected rescue team for itself and for the competing teams based on the proposed learning model that is presented in next section (III.B.(2)). As a result, we employ a model for integration of both neighborhood and environmental factors for giving the reward or penalty to the selected action. Then, the probability vector may changes based on the rewards and penalties parameters as a linear vector with assistance of learning automata with L_{ReP} scheme.

4. A key parameter in assigning special-temporal distributed tasks is time. Due to the importance of this parameter, we used a time period called time window for managing the time on the performance of the automata. Time window specifies the time duration in which an identified task can be assigned to the appropriate rescue team. Time window is calculated based on the criticality of different tasks. In other words, the defined time window supervises the learning duration of the automata in making decision on task assignment. When the time window 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 command center by the agent for making an optimal decision for the corresponding task.

5. In fact, the main objective of each automaton is to select a rescue team for allocating its task in the way that this selection is the best choice for that team as well as all the other search teams. 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.

2) The learning model used in the proposed method

After a specific cell selects an action a_i, the behavior of the cell is evaluated according to the following scheme:

$$\beta = T_1 \odot T_2 \odot T_3 \tag{3}$$

Where β is the response of the neighbors, T_1, T_2 and T_3 are three logical variables which are explained in Table 1, and $\odot$ represents the way that the logical variable are combined together to form the β.

Table 1. Description of logical variables for determining the response of the neighbors

	Condition
T_1	Does the selected rescue team have the same choice with its neighbors?
T_2	Is the selected rescue team located at the minimum distance from the competitor neighbors?
T_3	Is the selected rescue team a “critical choice” for the competitor neighbors?

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

$$\beta = \begin{cases} 1 & T_1 = T_2 = T_3 = \text{true} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

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 1, the concept of critical choice needs to be defined. The automaton investigates its next optimal choice and its neighbors’ next optimal choices using GIS functions and some important parameters like 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 critical choice for the neighbors. It means that if a competitor neighbor wants to neglect the team, it may lose some elements like time. Table 2 shows the model for combining the neighborhood and environmental factors.

Table 2. Model for integration neighborhood rule and environmental factors.

Neighborhood Rule	**Environmental Factors**	**Result for Selects *ai* 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

In the proposed model, environmental factor 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. Probability coefficients of Table 2 have been obtained by performing several tests and based on the fact that the automaton not converge quickly to the wrong action. The purpose of selecting these coefficients is evaluating of results of selected action by automata for 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 (compute β). Automaton gets penalty from its rival neighbors if this option is critical choice for even one of the competing teams. Thus, with the probability of penalty equal to 0.5, it evaluates this selection as the critical choice 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 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 time window is finished. Then, the task with priority 1 or 2 is assigned to the proper rescue team.

IV. Implementing the Geospatial Simulation Environment of Spatial Distributed Intelligent Agents

The following assumptions were made before implementing the proposed method:

- We first assumed that the number of critical points in each operational area and the estimated number of victims is collected by the loss estimator teams and they are located on the map.
- Without loss of generality, we assumed that the task allocation consist of search and rescuing.
- For the sake of simplicity, only major incidents relevant to the earthquake were taken into account, and secondary crisis such as aftershocks or subsequent fires are out of the scope of this study.

In order to study the effectiveness of our proposed model, Habibabad city was selected as the basis for simulations. Habibabad is one of the cities of Isfahan province, Iran, which is located in 10-Km northeast of Isfahan. The selected city is on the radius of the fracture and has been recently shaken with a strong 4.1-magnitude earthquake.

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 [30, 31].

ArcGIS Desktop software was used for designing desired GIS Shapefiles [44]. Moreover, DotSpatial as a geographic visualization tool of .NET was applied for working with GIS data [45]. Information about the area, the profile of the damaged buildings, specifications of search and rescue teams was entered to the database. To manage and access to the data and information by agents, we implemented a SQL database and used its functions. Fig.2 shows the schematic diagram of the implemented simulation environment. As shown in the figure, the GIS functions are used to add maps, layers and working with GIS data layers in the simulator. Model element represents functions to create the object of the search and rescue teams with behaviors, attributes and their respective functions on the map. Scheduler element represents the timing functions that its task is to manage the simultaneous implementation of search and rescue teams located on the map. Controller element also manages the activities of all the elements and shows other functions.

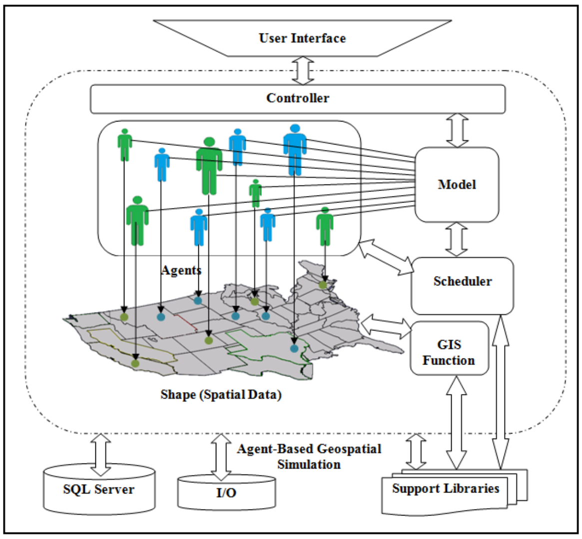

Fig.2 The schematic diagram of the implemented simulation environment

V. Simulating and Evaluating the Performance of Automata

A. *Output display of the program*

The final output of the designed program for geographic simulation of spatially distributed intelligent agents is given in Fig.3. As it can be observed in Fig.3, this program is an environment which shows the interaction of the human search team with its corresponding assistant agents. It also shows the interaction of the human rescue team with its relevant agent. For example, the search team with the unique characteristics of ID = 6 do the search task on the critical place with the unique characteristics of ID = 7. The situation of this place was considered as "Middle" considering the relief tasks that should be done. The number of identified injured individuals was equal to 6 individuals. The human user edits this information with the help of software agent in the (local and central) database and then requests its agent to assign the rescue task by clicking the “Task Allocation” button. Then, the decision-making process agent was implemented and the final decision was shown to its relevant human user. As it is demonstrated in the Fig.3, the search team agent selected the rescue team with the unique characteristic of ID = 5 for performing the relief tasks. Then, it sent the task to that selected agent and displayed the acceptance answer or rejection of the task by the human agent of the selected rescue team to its user. The human of the rescue team can also enter the new points identified by its agent into central and local databases by clicking the "Define New Point" button. New entered locations are also shown on the map. As shown in figure 3, new identified critical points along with detailed information are assigned to selected rescuer team as interdependent rescue tasks. In this simulation, we tried to show the adaption of our method with methods based on extreme teams [22] (teams which have four key characteristics “l, m, n, o” explained in II.B).

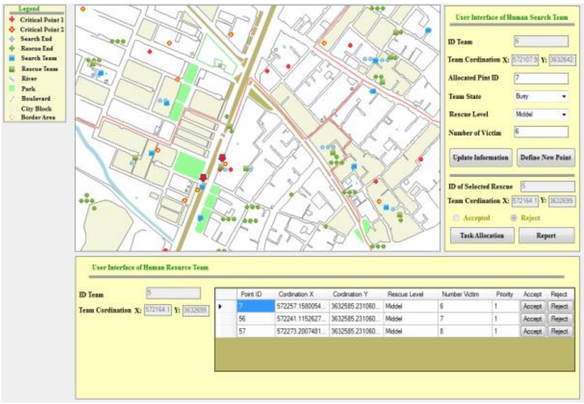

Fig. 3 Geospatial simulation environment

B. *The performance of designed automata of assistant agents of search teams*

In the diagram of Fig.4, the learning process of assistant agent for the search team with ID=1 is shown in the total time the scenario is executed. As can be seen from the diagram, agent related to search team selected an appropriate rescue team for assignment of each of the identified relief tasks. Given that the TimeWindow time monitors the automata performance, the time of selecting the appropriate rescue team by the search team for assignment of the relief task varies by the status of that task. For instance, the first relief task identified (T^1) was assigned to rescue team R_4, the second relief task identified (T^2) was assigned to rescue team R_1, the third relief task identified (T^3) was assigned to rescue team R_3 and the fourth relief task identified (T^4) was assigned to rescue team R_2 . As can be seen for assignment of the third identified relief task (T^3), at first the agent related to the search team selected the rescue team R_2. So, the agent gets penalty for this selection and reduces its probability because it was more critical to its neighbor. According to time management by TimeWindow on the automata performance, the agent selected the appropriate rescue team for assignment of the third identified task. So, the agent related to the search team with ID = 1 for assignment of each of the identified relief tasks selected the team with the highest probability.

VI. EXPERIMENTAL RESULTS

In order to evaluate our proposed model in accordance with the model used in the conventional network protocol (CNP) based on bid and auction from the reference [26] that was considered as the reference model, it is necessary to adjust similar time setting for both implementations. Therefore, in reference model, when the rescue team receives a task from the search team agent through a message, it calculates the price offered to do the requested relief task. This is done on the condition that the rescue team agent is able to do the task according to the level the relief task is requested. Then, it is placed on hold for 14 seconds. The search team agent also waits for the response from the rescue teams for 14 seconds after it sends the message to all rescue teams located in similar operating area. A number of different scenarios should be developed, so that each scenario may have its own parameters. The input parameters used for developing the scenarios are as follows: the number of rescue teams (N_{Rescue}) and the number of search teams (N_{Search}), the number of critical places, and the learning rate. The following graphs show the decrease in the total run time using the proposed model. In Fig.5 there are two scenarios, scenario 1 shows an earthquake with 50 critical places that each of places has different rescue level, status, number of injured people, and scenario 2 shows another earthquake with 100 critical places.

Fig.6 shows the evolution of the system performance index obtained with the cellular learning automata for task allocation of 100 critical points between rescue teams by different scenario of search numbers and rescue numbers:

(a) $L = \{N_{Search} = 15, N_{Rescue} = 15\}$

(b) $L = \{N_{Search} = 18, N_{Rescue} = 26\}$

(c) $L = \{N_{Search} = 20, N_{Rescue} = 15\}$

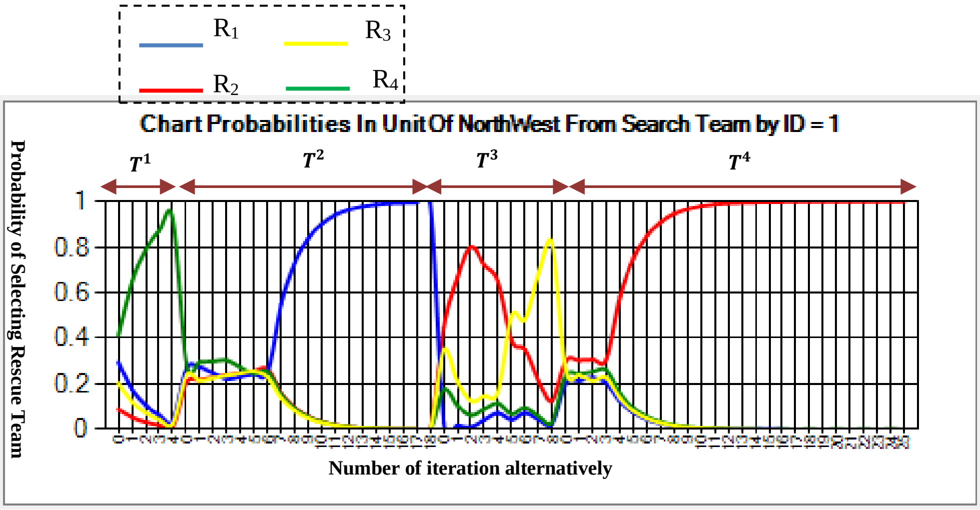

Fig. 4. the learning process of assistant agent from search team by ID=1 located in a operating area for assign rescue tasks of operation area to the rescue teams (R_1, R_2, R_3, R_4).

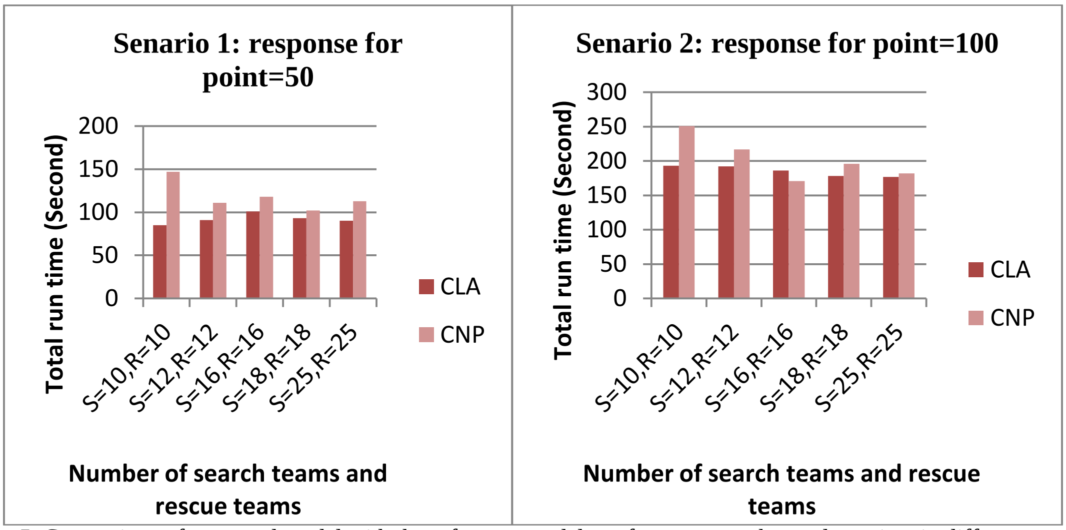

Fig. 5. Comparison of proposed model with the reference model: performance on the total run time in different states.

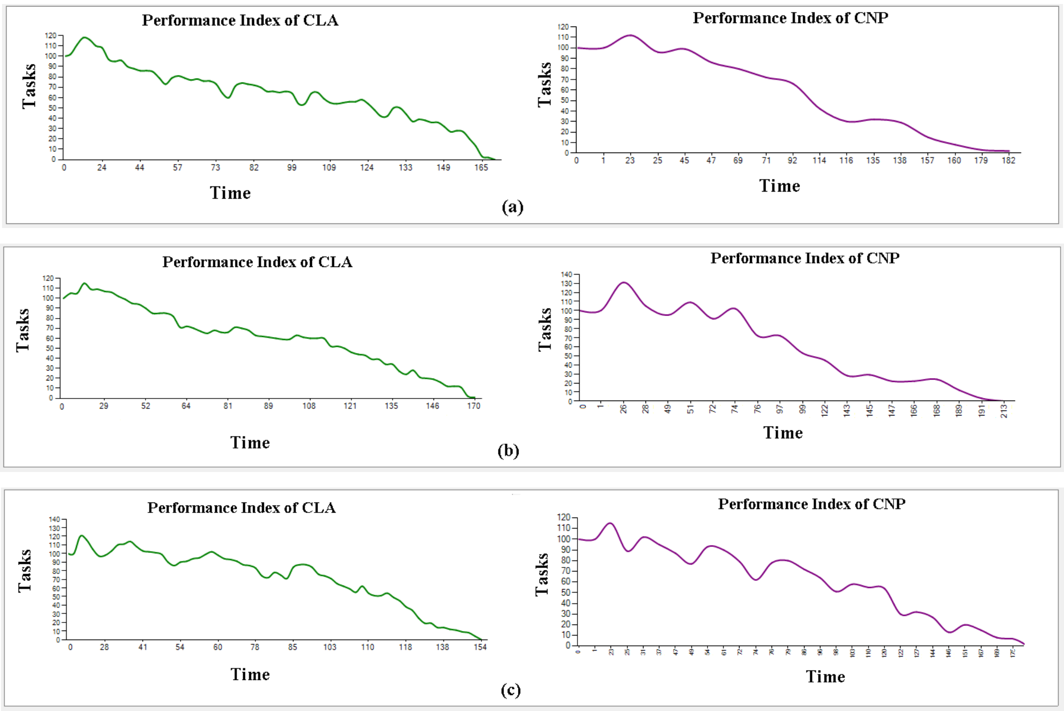

Fig6. Comparison of proposed model with the reference model: performance on the total run time in different states, with 100 critical points.

VII. Discussion and Conclusion

A key problem in urban search and rescue operations is spatial-temporal coordination between the team members. In order to solve this problem, an effective plan is the help of intelligent assistant agents, which are run on Personal Digital Assistant (PDA) devices and assist the human in making decision in emergency situations. Dynamic tasks identified by the human agent of the rescue team in the crisis environment. These tasks are characterized by spatial-temporal characteristics and are assigned to the appropriate rescue team by the intelligent assistant agents using intelligent decision-making techniques. This research aimed to study, propose, design and implement a new approach to improve the architecture of spatially distributed intelligent agents and to modify the multi-agent planning process structure. As the result, we were able to increase the learning ability of the intelligent assistant agents by using LA and add the interaction between the search agents to this architecture by using CLA. To evaluate the proposed algorithm, we designed a multi-agent geographical simulation and performed a number of different scenarios with a different number of rescue and search teams and critical points. As is clear from the results, we can have maximum utilization of the space and time in USAR operations. The evolution of the system shows performance index obtained with CLA for task allocation between rescue teams.

Finally, combining new learning automata with other learning methods can be a direction for future research. Moreover, making the simulation environment more realistic along with using geo-spatial information can be effective in obtaining better results.

References

[1] K.M. Khalil, M.H. Abdel-Aziz, M.T. Nazmy, and A.M. Salem, “Multi-Agent Crisis Response systems – Design Requirements and Analysis of Current Systems,” in: Fourth International Conference on Intelligent Computing and Information Systems, Cairo, Egypt, 2009, pp. 920-925.

[2] K.M. Khalil, M.H. Abdel-Aziz, M.T. Nazmy, and A.M. Salem, “The Role of Artificial Intelligence Technologies in Crisis Response,” Mendel Conference, 14th International Conference on Soft Computing, 2008.

[3] Rasekh A, Vafaeinezhad AR (2012) Developing a GIS Based Decision Support System for Resource Allocation in Earthquake Search and Rescue Operation. ICCSA 2012, Part II, LNCS 7334, Springer-Verlag Berlin Heidelberg, pp 275–285

[4] Adams, M. Field, E. Gelenbe, D. J. Hand. The Aladdin Project: Intelligent Agents for Disaster Management.

IARP/EURON Workshop on Robotics for RiskyInterventions and Environmental Surveillance, 2008.

[5] Ashish, Ronald Eguchi, Rajesh Hegde, et. al. (2007), “Situational Awareness Technologies for Disaster Response”, In Terrorism Informatics: Knowledge Management and Data Mining for Homeland Security, Ed:Hsinchun Chen et. al.

[6] Halliday. (2007), “Disaster Management, Ontology and the Semantic Web”, UCLA Department of Information Studies.

[7] Massaguer, Vidhya Balasubramanian, Sharad Mehrotra, and Nalini Venkatasubramanian. (2006), “MultiAgent Simulation of Disaster Response”, AAMAS’06 May 8–12 2006, Japan.

[8] Schoenharl, Greg Madey, G´abor Szab´o, et, al. (2006), “WIPER: A Multi-Agent System for Emergency Response”, Proceedings of the 3rd International ISCRAM Conference (B. Van de Walle and M. Turoff, eds.), Newark, NJ (USA).

[9] Schurr, J. Marecki, N. Kasinadhuni, M. Tambe and J.P. Lewis, Paul Scerri. (2005). “The DEFACTO System for Human Omnipresence to Coordinate Agent Teams: The Future of Disaster Response”, AAMAS’05, Utrecht, Netherlands.

[10] S. Jain and C. R. McLean. An Architecture for Modeling and Simulation of Emergency Response. Proceedings of the 2004 IIE Conference, 2004.

[11] Y. Murakami, K. Minami, T. Kawasoe, and T. Ishida. Multi-Agent Simulation for Crisis Management. KMN,2002

[12] M.H. Burstein, A.M. Mulvehill, and S. Deutsch, “An Approach to Mixed-Initiative Management of Heterogeneous Software Agent Teams, ” in HICSS, IEEE Computer Society, 1999.

[13] J.W. Crandall, C.W. Nielsen, and M.A. Goodrich, “Towards Predicting Robot Team Performance,” Systems, Man and Cybernetics, 2003, vol. 1, pp. 906- 911.

[14] D.V. Pynadath and M. Tambe, “Automated Teamwork Among Heterogeneous Software Agents and Humans,” Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS), 2003.

[15] P. Scerri, D. V. Pynadath, L. Johnson, P. Rosenbloom, N. Schurr, M. Si, and M. Tambe, “A Prototype Infrastructure for Distributed Robot-Agent Person Teams,” Autonomous Agents and Multiagent Systems (AAMAS), 2003, pp. 433-440.

[16] T. Fong, C. Thorpe, and C. Baur, “Multi-Robot Remote Driving With Collaborative Control,” IEEE Transactions on Industrial Electronics, 2003, vol. 50, pp. 699-704.

[17] G. Dorais, R. Bonasso, D. Kortenkamp, P. Pell, and D. Schreckenghost, “Adjustable Autonomy for Human-Centered Autonomous Systems on Mars,” In Mars, 1998.

[18] R. Hill, J. Gratch, S. Marsella, J. Rickel, W. Swartout, and D. Traum, “Virtual Humans in the Mission Rehearsal Exercise System,” KI Embodied Conversational Agents, 2003.

[19] N. Schurr, J. Marecki, J.P. Lewis, M. Tambe, and P. Scerri, “The DEFACTO System: Coordinating Human-Agent Teams for the Future of Disaster Response,” Published Articles of CREATE Research Archive, 2009.

[20] R.T. Maheswaran, C.M. Rogers, R. Sanchez, and P. Szekely, “Human-Agent Collaborative Optimization of Real-Time Distributed Dynamic Multi-Agent Coordination,” in: 9th International Conference on Autonomous Agents and Multi-agent Systems, 2010, pp. 49-56.

[21] Y. Quiñonez, J. de Lope, and D. Maravall, “Bio-inspired Decentralized Self-coordination Algorithms,” 4th International Work-Conference on the Interplay Between Natural and Artificial Computation (IWINAC), La Palma, Canary Islands, Springer-Verlag Berlin Heidelberg, 2011, pp. 30-39.

[22] F. dos Santos, A. L. C. Bazzan, “Towards Efficient Multiagent Task Allocation in the RoboCup Rescue a Biologically-Inspired Approach,” Autonomous Agents and Multi-Agent Systems, Springer, 2011, pp. 465-486.

[23] M. Koes, I. Nourbakhsh, and K. Sycara, “Heterogeneo Multirobot Coordination with Spatial and Temporal Constraints,” Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI), 2005, pp. 1292-1297.

[24] J. Godoy and M. Gini, “Task Allocation for Spatially and Temporally Distributed Tasks,” Proceedings of the 12th International Conference IAS-12, Springer-Verlag Berlin Heidelberg, 2013, Vol. 2, pp. 603-612.

[25] Y. Quiñonez, D. Maravall, and J. de Lope, “Stochastic Learning Automata for Self-Coordination in Heterogeneous Multi-Tasks Selection in Multi-Robot Systems,” 10th Mexican International Conference on Artificial Intelligence (MICAI), Puebla, Mexico, Springer-Verlag Berlin Heidelberg, 2011, pp. 443-453.

[26] R. Nourjou, M. Hatayama, and H. Tatano, ”Introduction to Spatially Distributed Intelligent Assistant Agents for Coordination of Human-Agent Teams’ Actions,” IEEE International Symposium on Safety, Security and Rescue Robotics Kyoto, Japan, 2011, pp. 251- 258.

[27] R. Nourjou and M. Hatayama, “Intelligent GIS for Spatial Cooperation of Earthquake Emergency Response,” Annuals of Disas. Prev. Res. Inst., Kyoto Univ., 2011.

[28] K. S. Narendra and M. A. L. Thathachar, ”Learning Automata: A Survey,” IEEE Transactions on Systems Man and Cybernetics, 1974, pp. 323‑334.

[29] A.H. Fathy Navid and A.B. Aghababa, “Cellular Learning Automata and Its Applications,” Computer and Information Science, Numerical Analysis and Scientific Computing, Publish with INTECH, 2013.

[30] A.T. Crooks and C.J.E. Castle, “The Integration of Agent-Based Modelling and Geographical Information for Geospatial Simulation,” Agent-Based Models of Geographical Systems, Springer, 2012, pp. 219-251.

[31] M. Fasli and M. Michalakopoulos, “Developing Software Agents using .NET,” University of Essex Department of Computer Science, 2006.

[32] S. Misra, K. I. Abraham, M. S. Obaidat, and P. V. Krishna, “LAID: A learning automata-based scheme for intrusion detection in wireless sensor networks,” Security Commun. Netw., Mar./Apr. 2009, vol. 2, no. 2, pp. 105-115,

[33] E. Fayyoumi and B. J. Oommen, “Achieving micro aggregation for secure statistical databases using fixed structure partitioning based learnin automata,” IEEE Trans. Syst., Man, Cybern. B, Cybern., Oct. 2009, vol. 39, no. 5, pp. 1192-1205.

[34] S. Misra and B. J. Oommen, “Using pursuit automata for estimating stable shortest paths in stochastic network environments,” Int. J. Commun. Syst., Apr. 2009, vol. 22, no. 4, pp. 441-468.

[35] V. Moghiss, M. R. Meybodi, M. Esnaashari, “An intelligent protocol to channel assignment in wireless sensor networks: Learning automata approach,” International Conference on Information Networking and Automation, 2010, pp. V1–338 - V1–343.

[36] S. Barzegar, M. Davoudpour, M. R. Meybodi, et al., “Formalized learning automata with adaptive fuzzy coloured Petri net; an application specific to managing traffic signals,” Scientia Iranica, 2011.

[37] J. Akbari Torkestani, “An adaptive learning automata-based ranking function discovery algorithm,” Journal of Intelligent Information Systems, 2012, vol. 39, pp. 441-459.

[38] M. R. Meybodi and M. R. Kharazmi, “Application of cellular learning automata to image processing,” J. Amirkabir, 2004, vol. 14, no. 56A, pp. 1101-1126.

[39] H. Beigy and M. R. Meybodi, “A self-organizing channel assignment algorithm: A cellular learning automata approach,” Springer-Verlag Lecture Notes in Computer Science, 2003, pp. 119-126.

[40] H. Beigy and M. R. Meybodi, “Asynchronous cellular learning automata,” Automatica, May 2008, vol. 44, no. 5, pp. 1350-1357.

[41] M. Esnaashari and M. R. Meybodi, ”A cellular learning automata based clustering algorithm for wireless sensor networks,” Sens Lett, 2008.

[42] M. Esnaashari and M. R Meybodi, “Dynamic point coverage problem in wireless sensor networks: a cellular learning automata approach,” Ad Hoc and Sensor Wireless Network, 2010, pp. 193-234.

[43] M. Talabeigi, R. Forsati, and M.R. Meybodi, “A hybrid web recommender system based on cellular learning automata,” IEEE international conference on granular computing, San Jose, California, 2010, pp. 453-458.

[44] M. Law and A. Collins, “Getting to Know ArcGIS for Desktop,” Environmental Systems Research Institute, 3nd editon,, 2013, pp. 19-30.

[45] A. Parks, “Using the Open Source DotSpatial GIS Library to Create Tasks for the Data for Environmental Modeling (D4EM) System,” Research Triangle Institute International, www.rti.org , 2011.

Maryam Khani Received her B.Sc. in Software Engineering from Mobarakeh Azad University of Esfahan, Iran and her M.Sc. in Mechatronic Engineering from Islamic Azad university, South Tehran Branch, Tehran, Iran. Her research interests include intelligent systems, human-agent interaction, multi-agent system and learning machine as well.

Ali Ahmadi Received his B.Sc. in Electrical Engineering from Amir kabir University, Tehran, Iran in 1991 and his M.Sc. and Ph.D. degree in Artificial Intelligence and Soft Computing from Osaka Prefecture University, Japan in 2001 and 2004, respectively. He worked as a researcher in Research Center for Nano devices and Systems in Hiroshima University, Japan during 2004–2007. He has been with K.N. Toosi University of Technology, Tehran, Iran as assistant professor from 2007. His research interests include distributed intelligent systems, human-computer interaction, computational intelligence, adaptation and interactive learning models, virtual reality and artificial life as well.

Maryam Khademi Received her B.Sc. in Mathematics from Kharazmi University,Tehran in 1989 and her M.Sc. in Pure Mathematics from Islamic Azad University North Tehran Branch in 1993 , and another M.Sc. and her Ph.D. in Applied Mathematics respectively from Iran University of Science and Technology University and University of Tehran in 1994 and 2004.She has been working as assistant professor at Islamic Azad University South Tehran Branch since 1995.Her research interests include Computer Science, Computational intelligence ,Data mining ,Optimization problems and E-learning.

IJICT

This Page intentionally left blank.