Integrated Genetic Algorithmic and Fuzzy Logic Approach for Decision Making of Police Force Agents in Rescue Simulation Environment

Ashkan Radmand, Eslam Nazemi, and Mohammad Goodarzi

Electrical and Computer Engineering Department, Shahid Beheshti University, Tehran, Iran
{a.radmand,nazemi,mo.goodarzi}@sbu.ac.ir

Abstract. The major task of police force agents in rescue simulation environment is to connect the isolated parts of the city. To achieve this goal, the best blocked roads should be chosen to clear. This selection is based on some issues such as number of burning buildings and victims existing in the mentioned parts. A linear combination of these factors is essential to determine a priority for each road. In this paper we propose an integrated Genetic Algorithm (GA) and Fuzzy Logic approach to optimize the combination statement. The parameters are learned via GA for some training maps. Then, because of differences between test and train maps, the agent should decide which parameters to choose according to the new map. The agents' decision is based on similarity measures between characteristics of maps using Fuzzy Logic. After utilizing this method, the simulation score increased between 2% and 7% in 20 test maps.

Keywords: Rescue Simulation, Police Force Agent, Decision Making, Genetic Algorithm, Fuzzy Logic.

1 Introduction

In rescue simulation environment, a simulated earthquake happens and the city goes in an emergency state. Some buildings start to burn, some others collapse, some civilians get damaged and blocked in collapsed buildings and some city roads close by debris caused by disaster. These blocked roads divide the city roads graph into isolated city parts. The major task of police force agents [1] is to connect the isolated parts of the city that causes the easier transportation of other types of agents (Fire brigades and Ambulances) to rescue the city and its civilians. To achieve this goal, the best blocked roads should be chosen to clear. This selection is based on some issues such as number of refuges, stuck agents, burning buildings and victim civilians existing in the mentioned parts.

Therefore, the police force agents should decide which two city parts are more important to get connected first. A linear combination of these factors is essential to determine a weight (priority) for each road.

J. Baltes et al. (Eds.): RoboCup 2009, LNAI 5949, pp. 288–295, 2010.

In section 2, we will explain major strategy of police force agents to build a linear combination of decision factors which were mentioned above. Section 3 is about description of the GA approach which has been used to learn the optimum parameters of police force agents' decision making in some fixed training maps. Section 4 explains the method of combination of the trained parameters to achieve an efficient solution for unknown city maps using Fuzzy Logic. We utilize fuzzy logic to determine the measure of similarity between new maps and learnt maps based on some characteristics of them such as state of fires, blockades, victim civilians and etc. Finally in section 5, some experimental results are reported to show the effect of proposed method on agents work efficiency.

2 Police Force Agents Main Strategy to Choose a Target Road

Police force agents should connect all city parts together as soon as possible and in a manner that leads to a higher final score which determines the performance of agents work. To achieve this goal the agents have to assign a weight to each boundary road (means any road that disconnects a city part from another) and start to open them based on these weights or priorities.

The police agents consider some conditions which determine the worth of cleaning each boundary road. The final weight of that road equals to the sum of these conditions values. The value of a condition is calculated based on its premise parameters. Each condition value should get a weight in summation step of calculating final weight of a boundary road. These weights determine the importance of each condition comparing to the other conditions.

In order to better understand, assume that we have a boundary road *l* which separates two city parts *cp1* and *cp2*. Considering this situation, we have conditions described in Table1.

Table 1. Conditions to assign a weight to a boundary road

Condition	If part #1	If part#2	If part#3	Then part
Cond. 1	#(cp1.BB)>0	#(cp1.RF)=0	#(cp2.RF)>0	#(cp1.BB)* W1
Cond. 2	#(cp1.BB)>0	#(cp1.FB)=0	#(cp2.FB)>0	#(cp1.BB)* W2
Cond. 3	#(cp1.DV)>0	#(cp1.AT)=0	#(cp2.AT)>0	#(cp1.DV)* W3
Cond. 4	#(cp1.DV)>0	#(cp1.RF)=0	#(cp2.RF)>0	#(cp1.DV)* W4
Cond. 5	#(cp1.BA)>0	#(cp1.AT)=0	#(cp2.AT)>0	#(cp1.BA)* W5
Cond. 6	#(cp1.BF)>0	#(cp1.AT)=0	#(cp2.AT)>0	#(cp1.BF)* W6
Cond. 7	#(cp1.BP)>0	#(cp1.AT)=0	#(cp2.AT)>0	#(cp1.BP)* W7

BB: Burning Buildings RF: Refuges DV: Dying Victims
AT: Ambulance Teams FB: Fire Brigades PF: Police Forces
BA: Buried* AT BF: Buried FB BP: Buried PF
* A buried agent is a victim agent that needs ambulance team help.

Each condition consists of three “if parts” combined with AND operator and one “then part” affected by a coefficient *Wi* which is the weight of the condition in summation step. For example, the first condition says that "if the number of burning buildings in *cp1* is more than zero AND the number of refuges in that city part is equal to zero AND the number of refuges in *cp2* is more than zero", then the condition 1 has a value equal to number of burning buildings multiplied by *W1*. The coefficient *Wi* is the weight of i-th condition in calculating the total weight of *l*.

The importance of connecting two city parts, one with burning buildings and other with refuges, referring to the first condition, is determined by the fact that fire brigades need to go to the refuges to fill their water tanks in order to extinguish burning buildings.

Consequently, decision making of police force agents depends on *Wi*. To achieve good weights for conditions we have used a GA approach which will explain in section 3.

3 Genetic Algorithm Approach to Determine Weights

In section 2, we described the police force agents' general strategy to choose target roads. To optimize the performance, they should assign good values to conditions weights. To achieve this goal, we utilize GA approach [3-5] that will be described in this section.

3.1 Chromosome Structure

In our method, chromosomes have a simple structure which is an array of values assigned to weights of conditions. An example of chromosomes structure has been shown in Fig1.

30	40	60	200	150	250	100
1	2	3	4	5	6	7

Fig. 1. An example of the chromosome structure

The size of each chromosome is equal to *n* which represents the number of conditions mentioned in section 2.

Initial population may be generated through a random or user specified process. It plays an important role in search direction. A well selected initial population increases the search procedure convergence speed and results in faster trend to optimum solution. In the proposed method, to generate initial population, values assigned to chromosomes are the same as values used before by experiment. Only one element or two of each individual chromosome take a random value.

After constructing initial population, the fitness values for all individuals should be calculated. The number of individuals in the population is constant in all generations. Some individuals that have most fitness values are gone forward to next generation. If the crossover rate is called *Pc* and number of individuals is called *Ps*, number of individuals that are passed to next generation is equal to *Ps- Ps*Pc*. Therefore, the number of new generated individuals in each generation is *Ps*Pc*. These processes are performed while the terminating condition is not satisfied. Other parts of the proposed genetic algorithm will be described in sections 3.2, 3.3 and 3.4.

3.2 Fitness Function

In the presented method, the total score of simulation is used as fitness value. This score is calculated based on work efficiency of all agents. All factors of simulation including city map, initial fires, victim civilians and decision making algorithm of other agent types (fire brigades and ambulances) should be fixed in GA training iterations. The only exception is police forces work which differs in the various training iterations. This difference is because of the changes occurred in the weights in the decision making section. So, any change in the total score is because of the change in work of police force team.

3.3 Selection Algorithm

The selection of the individuals is based on the fitness value of the solutions. The probability of selection of an individual is directly or inversely proportional to its fitness value. The roulette wheel selection [6] is used in our proposed GA. The main idea of this method is to select individuals stochastically from one generation to create the next generation. In this process, the more appropriate individuals have more chance to survive and go forward to the next generation. However, the weaker individuals will also have a little probability to select.

In selection process, *Ps*Pc* individuals are selected to create the same number individuals from them using crossover and mutation operators.

3.4 Crossover and Mutation Operators

Since individual chromosomes based on a simple structure, complex cross over operators are not necessary. In the proposed method, two point crossovers are used. Therefore, *Ps*Pc* individuals are selected using our selection process where *Pc* is crossover rate. As *Ps*Pc* new individual is needed after doing crossover, two parents are selected and two new child are produced from them. Points in each parent are selected randomly and segments between these two points are substituted by parents to produce new individual children.

We produce a random number for each element of individual chromosomes. If it is lower than *Pm*, mutation will be done for that element. Note that *Pm* is mutation constant. Three elements of each individual chromosome can be chosen for mutation at most. Terminating criteria is the number of generations which is determined.

4 Decision Combination Using Fuzzy Logic

In rescue simulation environment, each map has some initial states such as start points of fire spread, victim civilians' positions, blocked roads and etc that affect the decision making strategy of police force agents. Therefore, it is not applicable to learn conditions weights in a certain map and use them in any other map. On the other hand, we cannot learn the weights for all possible maps because the number of maps is infinite.

To overcome this problem, we designed some training maps with specific characteristics. We have used some characteristics that are more important in classifying the maps including number of burning buildings, number of victim civilians, number of blocked roads and number of buried agents.

Each of these maps has different values of above characteristics. In fact, training maps are representatives of all maps. Applying GA approach mentioned in section 3 to all training maps gives us an array of optimum weights for each map. It is required to have a method to combine these arrays and have a solution for each new map. In this section, we will propose a method of decision combination using fuzzy logic [7-9].

To achieve this goal, each characteristic of map will create a fuzzy set that consists of three membership functions: *Low*, *Medium* and *High*. Fig. 2 demonstrates the membership functions of maps characteristics.

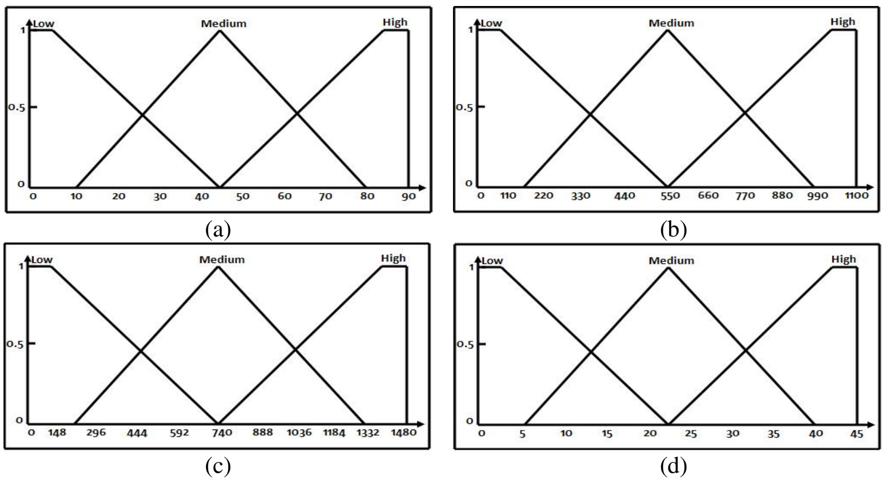

Fig. 2. "(a) Victim Civilians number (b) Burning Buildings number (c) Blocked Roads number (d) Buried Agents number" Membership Functions

4.1 Fuzzy If-Then Rules

Using defined fuzzy sets, fuzzy if-then rules will be created. Each rule has four "if parts" combined with AND operator pointing to map characteristics mentioned above and one "then part" that relates the given map to one of training maps. These rules are in Mamdani's proposed form [10]. For example, one of the rules is:

IF "number of victims" is High AND "number of burning buildings" is Low AND "number of blocked roads" is Low AND "number of buried agents" is Medium THEN map IS training map #1.

Therefore, to cover all possible conditions, we should create $3^4 = 81$ rules. So, we should design 81 training maps that each one has characteristics as same as those which are described in the center of fuzzy membership functions of corresponding rule. Having these rules, the training part is completed and we just need a fuzzy inference method that can estimate suitable weights for any new given map. This fuzzy inference method is explained in section 4.1.

4.2 Fuzzy Inference Method

Given a new map, values of map characteristics will be checked against all fuzzy If-Then rules. Each "if part" of a rule has a membership function. So, the membership value of the given map will be calculated for each of them. As a result, for a given map there will be four membership values. A rule "trigger value" (means rule weight in fuzzy inference) will be equal to minimum membership value of "if parts" of that rule.

In defuzzification step, to achieve final weights for the given map, we should combine "then part" of all rules considering their trigger values. "Weighted Average" method is used in our proposed method for defuzzification step.

For example, assume that we have just two fuzzy rules. After checking the rules for a given map, the first rule has gotten a trigger value *w1* and refers to training map #1 in its "then part", and the second rule has gotten a trigger value *w2* and refers to training map #2 in its "then part". So, the final weights will be calculated based on equation (1).

$$final\, weights = \frac{w1 \times weights1 + w2 \times weights2}{w1 + w2} \tag{1}$$

weights1, weights2 and final weights are the weights arrays of training map #1, training map #2 and the given map respectively.

5 Experimental Results

In order to evaluate the presented algorithm, we implemented it on "SBCe_Saviour2008" source code which was one of the eight top teams in China2008 competitions. In this section, to demonstrate the performance of the method, we compared the results of our method with the results gained before its implementation. 81 maps with different characteristics were designed for training section. The mentioned GA-based method was applied to each map. Using try and error, GA generations consist of 10 chromosomes and the learning process continued for 100 generations for each training map. Crossover and Mutation probabilities were set to 0.8 and 0.2 respectively. Change in value of elements, which had been mutated, was equal to ±20 in first generation and reduced in a way that it reached zero in 10 last generations. Fig. 3 demonstrates changes in maximum gained scores in each generation during GA training step. All the training maps were based on Foligno city which is one of the standard maps in rescue simulation league of RoboCup competitions [2].

Maximum total score is increased about 7% as it is shown. In other 80 training maps this increase was between 5% and 8%. After completion of GA training step for all maps, 81 fuzzy if-then rules were created and each one was assigned to a training map in its "then part".

20 test maps were chosen from Robocup2009 China rescue simulation competition maps. The comparison of total scores gained using proposed method and the results gained before the training is shown in Table2.

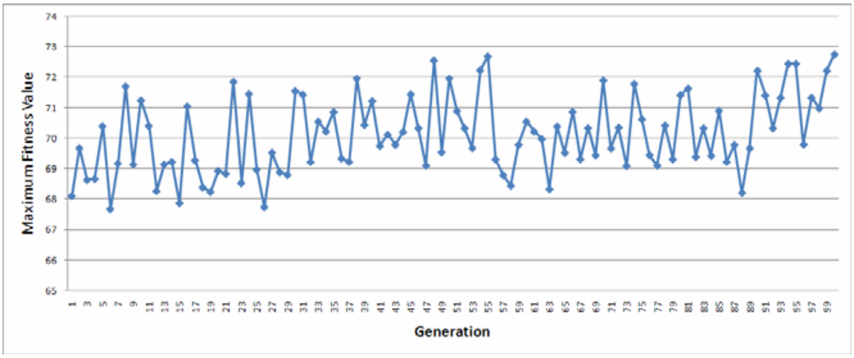

Fig. 3. Maximum gained scores in training generations

Table 2. The comparison of results gained before and after training

Map Name	Old score	New score	Map Name	Old score	New score
Map #01	65.356	67.64	Map #11	90.012	92.103
Map #02	63.165	67.265	Map #12	93.464	95.689
Map #03	76.798	78.013	Map #13	88.645	90.465
Map #04	85.679	88.625	Map #14	87.465	91.465
Map #05	72.946	75.346	Map #15	85.856	87.695
Map #06	59.689	62.345	Map #16	75.964	78.654
Map #07	75.899	78.463	Map #17	72.331	74.649
Map #08	77.689	79.341	Map #18	70.002	73.566
Map #09	74.334	76.555	Map #19	90.645	92.256
Map #10	82.645	85.756	Map #20	81.135	84.698

6 Conclusion

In this paper we proposed a GA-based approach to achieve suitable weights for decision making conditions of police force agents in rescue simulation environment. This method is applied to some training maps and the result weights will be combined to achieve suitable weights for any new given map using fuzzy inference method. The simulation results showed that the method which was presented has positive effect on decision making of police force agents and increases the total score of whole team.

References

1. Morimoto, T.: How to Develop a RoboCup Rescue Agent. Rescue Technical Committee
2. RoboCup Rescue official Website, `http://www.robocuprescue.org`
3. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
4. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Kluwer Academic Publishers, Boston (1989)
5. Banzhaf, W., Nordin, P., Keller, R., Francone, F.: Genetic Programming - An Introduction. Morgan Kaufmann, San Francisco (1998)

6. Al Jadaan, O., Rajamani, L.: Improved Selection Operator for GA. Journal of Theoretical and Applied Information Technology, 269–277 (2005)
7. Kruse, R., Gebhardt, J., Klawonn, F.: Foundations of Fuzzy Systems. John Wiley & Sons, New York (1994)
8. Dubois, D., Prade, H.: Fuzzy Sets and Systems: Theory and Applications. Academic Press, New York (1980)
9. Zimmermann, H.-J.: Fuzzy Set Theory—and its Applications, 2nd edn. Kluwer Academic Publishers, Boston (1991)
10. Mamdami, E.H., Assilina, S.: An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies 7(1), 1–13 (1975)