© 2015,
Scienceline Publication
www.science-line.com

ISSN: 2322-5114
Journal of World’s Electrical Engineering and Technology
J. World. Elect. Eng. Tech. 4(1): 01-06, April 25, 2015

JWEET

Path-Finding in Multi-Agent, Unexplored and Dynamic Military Environment Using Genetic Algorithm

Saeed Saeedvand*, Seyed Naser Razavi and Fahimeh Ansaroudi

Computer Engineering Department, Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran

*Corresponding author's Email: S-Saeedvand91@ms.tabrizu.ac.ir

ORIGINAL ARTICLE
PII: S232251141500001-4
Received 03 Aug. 2014
Accepted 22 Sep. 2014

Abstract *–Path-finding in multi-agent, unexplored and dynamic military environment is one of the most important issues for solving in simulators. In this article a substructure of agent Path-finding is created which these agents are placed in an unexplored environment, also they communicate each other and use proposed methods to find targets to reach them. The most important aim of this article is to consider the targets behavior to decrease the required cost for reaching to the targets by improving agent’s behavior in a dynamic military environment and improving agent’s interactions. In the proposed method each agent has movement ability in a dynamic environment autonomously and they have ability to finding targets between obstacles. Therefore to solve this problem all of the necessary constraints to find path in a dynamic and unexplored environment are considered and Genetic algorithm is used.*

Keywords: *Multi-Agent System, Path-finding, Chromosome, Fitness Function*

INTRODUCTION

In the most different sciences path-finding one of the most important challenges which have attracted researchers is finding a solution by considering more factors and different constraints [1-4]. In a military environment, agents usually interact with a dynamic environment, so for reach different targets and find desired results, they have to corporate dynamically with each other. In a dynamic environment different factors are involved in each agent’s behavior which cause to a NP-Complete problem. Each agent in an environment has some potential such as range of vision, movement speed, depth of view, the ability to communicate other agents, and so on. In such environments agents are not aware of path quality, before testing it and in most cases they should test it to determine this. The aim of this research is to create a substructure considering some of constraints and potentials in each agent to find a path which this substructure can be generalized, improved and combined to consider more constraints and potentials. Since solving large state space problems like this are NP-Complete and Genetic algorithm is one of the best searching tools in such problems [5].

Faez Ahmed [6] presented Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithms. In the proposed genetic algorithm to find the shortest path to reach target, the length of the path was minimized while path safety and path smoothness was maximized. It is noteworthy that agents are aware of exact location and coordinate of targets by default. To solve this problem, genetic algorithm with elitist non-dominated domination was used. Also, they mentioned that other evolutionary methods can be used to solve this problem. Simultaneously, they presented solutions to multi-agent problem in large space with some obstacles existing there. Shin-Jin Kang [7] presented live path: adaptive agent navigation in the interactive virtual world. In this work a new method is presented to find a path for agents in virtual world in which they used obtained data from user’s behavior and also the techniques such as Agent of Interest (AOI), Region of Interest (ROI) and Discredited Path Graph (DPG). They used user’s decisions and preferences in smart path-finding in their method. Since in the real world, unexplored areas do not have roads, to find path in such areas natural and instinctive behavior is needed. Huang Jin [8] Presented Multi-Agent Path-finding System Implemented on XNA. XNA is a new game development framework based on DirectX released by Microsoft [9]. In this method each agent thinks quickly and moves to reach its destination. They presented the idea of collision avoidance between obstacles and between agents and they expressed that by solving this problem many problems in real-life applications, including motion planning in robotics, air traffic control, vehicle routing, disaster rescue and computer games will be solved.

Agents path finding challenges

Agent path finding in a military environment involves different items. One of the most important challenges in a military environment is due to dynamic environment which varies with time. In such environments the problem is that agents cannot find their path to reach a definite

To cite this paper: Saeedvand S, Naser Razavi S and Ansaroudi F. 2015. Path-Finding in Multi-Agent, Unexplored and Dynamic Military Environment Using Genetic Algorithm. *J World Elec. Eng. Tech.*, 4 (1): 01-06.
Journal homepages: http://www.jweet.science-line.com/

target by using common methods [10]. The reason is due to strategic impression of the environment on the problem. As we know an agent can be a tank, a lift, a soldier and any movable agents. When an agent is moving toward a specific target, some factors such as position of relative forces, position of enemy's forces, the strategies of forces and so on are impressive in path-finding. So, it is not possible to use common path-finding methods to reach targets. In a military environment each agent has its own unique characteristics which give them some capabilities. These characteristics include all the interactions which an agent can perform in an environment or all the information which an agent can obtain from the environment and share them between other agents. Due to this problem's large state of space, in this work we attempt to design a simulator which can create an extensible infrastructure for agent path-finding in 2d environment. This is the first stage of the simulation which other factors can be imposed in performance of agents without any changes in the proposed method.

Genetic algorithm and a history of genetic algorithm

Genetic algorithm was first introduced by John Holland [11]. It is a searching method in computer science field to find a heuristic solution for optimization and search problems with large state of space. Genetic algorithm is a kind of evolutionary algorithms which uses biology methods such as inheritance and mutation. In fact, genetic algorithms use the principles of Darwinian natural selection to find optimum formula to predict pattern matching. Generally, genetic algorithm is an appropriate method for searching in large state of spaces which the result is not necessarily the best one but it is near optimum.

Assumptions and formulation

In this section some assumptions are considered to solve agent path-finding problem. These assumptions are as following.

- Fixed and limited numbers of agents are in environment (A).
- Each agent has a limited depth of view (R).
- There are limited numbers of targets in environment (T).
- Depth of view of each agent is considered circularly, in fact if target is a person then 360 degree is considered to head rotation and visibility.
- Each agent has capability of viewing other agents and targets which are in its depth of view.
- Movement speed of agents is considered to be homogenously.
- Agents have capability of either making communication between each other or informing their own position e.g. by using wireless and Global Positioning System (GPS).
- Each agent has capability of moving toward four main direction and four secondary direction (Figure 1).
- The environment is considered as a 2D plane (x, y).
- (z, w) is the coordinate point which each agent starts to move from and its equal with (x, y) in the start time.
- Each agent in every time (t) can move one step in the coordination of environment.
- In environment there are some obstacles with different sizes and coordinates which agents cannot pass them
- Each time only one agent can be placed in position of environment. (Agents are not able to pass each other).
- Each agent is able to move one direction in the size of its depth of view.
- All agents in one generation can move in distinct and predicted number of steps (L).
- Each agent in one step cannot return to its previous position. (The previous direction is stored and in new step agent is not able to move in reverse).
- When the target is in the agent's depth of view then agent traverse one straight line or direction to reach the target.

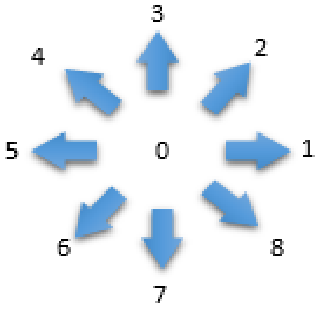


Figure 1. The Agents movement form and directions.

Agents movement strategy

When each agent is follows some targets in environment and considers some cases which determines the agent's strategy. These cases are as follow.

1. Each agent moves in size of its depth of view. In other words according to the problems assumptions when an agent is in assumptive coordinate (x, y) can view the environment in the size of determined depth of view. So, it moves in the size of its depth of view and searches the environment.
2. Each agent while moving in environment considers other agents and obstacles to avoid collision in its path.
3. The environment is dynamic and changes all the time so behavior of agents should be changed dynamically in the problem.

Genetic algorithm and agents path-finding chromosome structure

In the section of the algorithms chromosome structure it should be declared that this structure is considered as a

2 dimensional array so that, movement path of each agent is stored in the array with values of (x, y) in 2 dimensional coordinate. In the proposed method all the procedures of genetic algorithm are implemented as Figure 2.

MATERIAL AND METHODS

The proposed genetic algorithm

In this work it is assumed that agents are not aware of the targets position. And by using the presented methods try to first, find a path to reach it by their own knowledge and secondly improve the path found by them. In the next sections every steps of this phase will be declared separately.

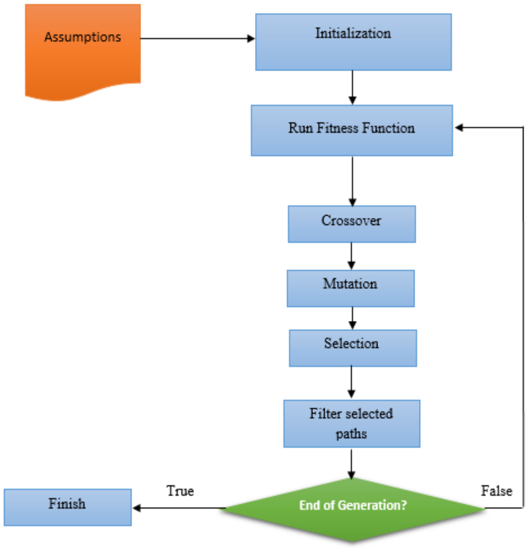


Figure 2. The flow chart of the proposed algorithm for agent path-finding.

A. Initialization

In this step initial chromosomes population will be created. Chromosomes population in agent path-finding section consists of some agents for path-finding. These agents considered in one range with predefined coordinate in environment. Agents placement in environment is so that, one random starting point (z, w) in determined range is created and forms the starting point of the agents movement path. In the next step, movement path of each agent according to starting point and described assumptions in previous section is an array of point (x, y). This movement path in initialization step is determined randomly and is a trail of movement directions which is determined in the path that agents can move.

B. Mutation Operators

In the proposed method the idea of mutation in the chromosomes is in three different ways which are performed using determined rates to select some chromosomes and have mutation on them.

1. **Mutation type one:** Repetitive paths in the agent's movement path will be eliminated. In other words, in this step if an agent starts from one point at time t and return to this point at time (t + n) this path which has created a ring will be eliminated from the chromosome. After this elimination a new path in the size of the length of eliminated steps will be added to chromosome randomly (Figure 3).

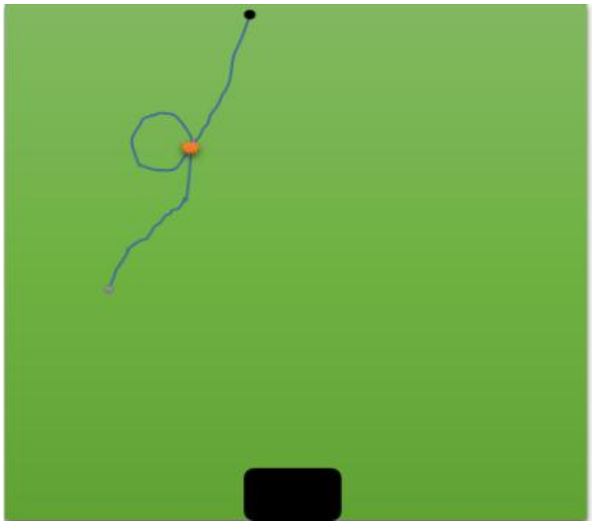


Figure 3. Mutation type one, additional movement path of an agent.

2. **Mutation type two:** In mutation type two a value smaller than the length of chromosome is considered as mutation rate randomly to have mutation on chromosome, and then using resulted direction array the chromosomes remaining path will be rewritten.

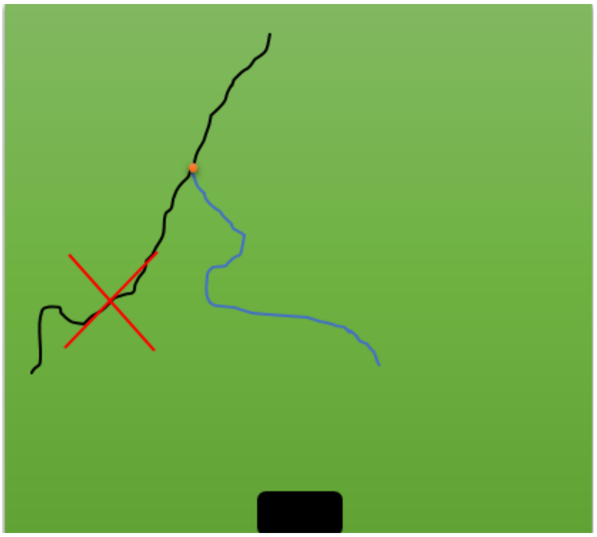


Figure 4. The Effect of Mutation Type Two in Agents Movement.

3. **Mutation type three:** Mutation type three will only be applied on chromosomes which have found target before. In this mutation a part of the agent's path will be replaced with a new path (Figure 5).

The aim of this path replacement is to minimize the agents movement steps to reach the target and will be as follows. First two random points will be selected in chromosome and then a straight line will be drawn between two points using equation of line. This line will be drawn when there is no obstacle between two points in environment. The coordinates obtained from drawing the line will be replaced with the previous coordinates in chromosome. Now the length of new created path is smaller or equal to the previous path. So, first the coordinates in chromosome will be shifted and then the remaining path will be rewritten by creating random paths.

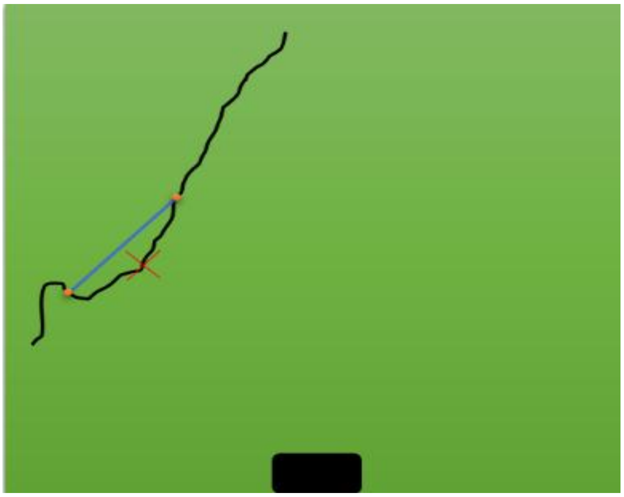


Figure 5. The effect of mutation type three in movement of the agent

C. Crossover Operation

The proposed operator for crossover is selecting two agents randomly from parent chromosomes population and then applying two-point crossover on chromosomes. Procedures of applying crossover operator are shown below.

1- Selecting two parents randomly from the population of parent chromosomes.

2- Converting selected chromosomes to an array of movement directions.

3- Applying two-point crossover on array of movement directions.

4- Converting the array of combined movement directions to a form of chromosomes with 2 dimensional coordinates.

In crossover operator it will be noticed that when structure of the chromosome demands high computational cost, this operator will be ignored and the best path to reach targets will be obtained by mutation operator.

D. Fitness Function

In the second phase, to compute fitness function according to determined targets for agents, in each agent the last obtained movement coordinate is considered and then the Euclidean distance from these coordinates to the targets coordinates will be computed. After obtaining this distance, the number of steps which the agent has traversed to reach the target will be added to fitness value. So, the obtained value is as fitness value in chromosome which a small value for it indicates that how much this chromosome is good. Equation (1) indicates the fitness function in second phase.

$$F_i = \left(\sqrt{(x - xT)^2 + (y - yT)^2}\right) + (Si) \qquad (1)$$

In the above equation si indicates the number of steps which the i'th agent has traversed to reach determined target. The xT and yT values also indicate the targets current coordinate.

A. Selection Function

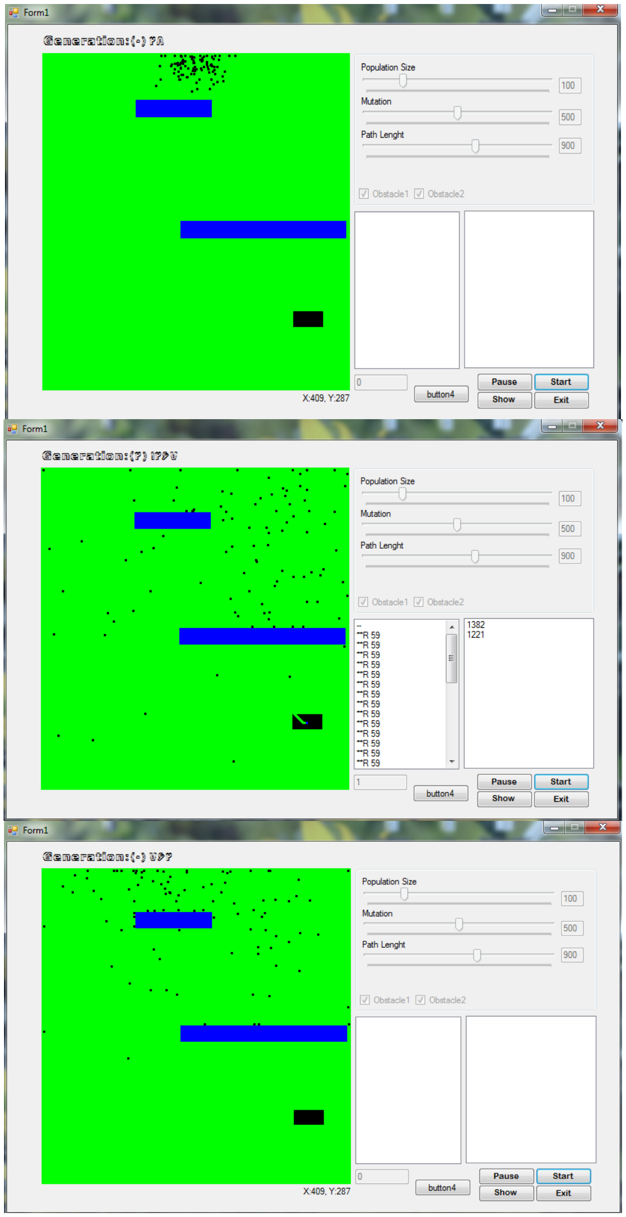
In this method for reselection of initial population to implement in next generation, child population based on fitness value will be arranged descending. Then using elitism, 10 percent of population will be transferred to new parent population then by using roulette wheel, remained parent population will be selected [12]. In this way selecting chromosomes with higher fitness value is more probable than selecting chromosomes with lower fitness value.

RESULTS

To implement agent path-finding algorithm we have used C sharp programming language 2013 and Direct X GUI version 10 (Figure 6). In the presented algorithm for running simulations and getting results, in addition to explained assumptions in chapter 4 the following assumptions are considered, too.

- Agents are placed in a two dimensional environment with the size of 420 × 420.
- The agents start point is a random point in coordinate range 20 × 20 from top of environment.
- In experiments the target which agents follow it, is placed in a random point in bottom of environment and far from starting point.
- The agent's movement speed in environment is considered pixel by pixel.
- A number of obstacles in different sizes and dimensions are placed in environment.

Figure 6 indicates the steps of implementation and movement of agents in an environment. In future, the results of implementation of genetic algorithm in four different ways are indicated. Considered different parameters for each type are shown in (Table 1).



Population Size 100
Mutation 500
Path Lenght 900
☑ Obstacle1 ☑ Obstacle2

Figure 6. Implementation of designed two dimensional simulator in existence of obstacles in environment and movement of 100 agents through targets in the first phase of the algorithm

Table 1. Values for implementing algorithm in simulator

	Mutation	Path Length	Population size	Obstacle
GA1	10%	800	1000	**False**
GA2	15%	800	1000	**False**
GA3+(OBS)	10%	800	1000	**True**
GA4+(OBS)	15%	800	1000	**True**

In Table 1, the cases such as mutation rate, length of the agent's movement path in each generation, number of initial population and existence of a number of obstacles in difficult condition are specified.

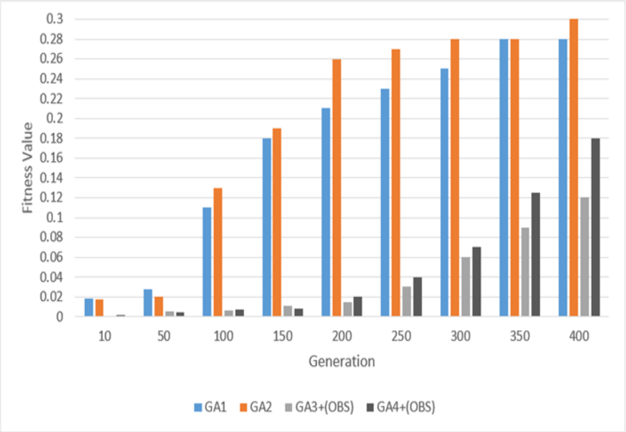


Figure 7. Results obtained from different implementation of the proposed algorithm.

Simulation results indicate that average fitness value for agents in the absence of obstacles in environment in both GA1 and GA2 after maximum 350 generation have reached up to 0.28. In fact, the value 0.28 indicates the average quality of found path by all agents in relevant generation. But, in existence of obstacles in environment which GA3 and GA4 indicate this, the fitness value after 400 generation reaches up to 0.1. The value 0.1 indicates that most of agents in that generation have reached target from short and appropriate path. The point is that in the absence of obstacles the algorithm converges to optimal solution very quickly but in existence of obstacles in environment the speed of convergence is low. In fact, the presented algorithm in 400 generation reaches a good solution which is a good number in genetic algorithm.

CONCLUSION

In the proposed research a number of agents were considered in a dynamic environment. In this research a substructure was created for agent path-finding so that, agents are in unexplored environment and they reach the desired targets by communicating each other and using the proposed methods. The most important aim of this research was to considering the targets behavior so that, we can improve the value of cost to reach targets, by improving the agent's behavior in a dynamic military environment and by making communication between this agents. The main reason of using genetic algorithm is due to multi-agent environment and also other factors involved in the problem. Implementation results indicate that by considering constraints and assumptions, genetic algorithm can create the best paths to reach targets in a few numbers of generations. This makes it possible to add other factors in military environment for agents decisions in the future works.

REFERENCE

[1] N. J. E. R. Yun-Wu Huang, "A Hierarchical Path View Model for Path Finding in Intelligent Transportation Systems," GeoInformatica , pp. 125-159, 1997.

[2] L. Shelton, "Optimizing modern path finding methods In Imperfect 2D Environments," Journal of Game Behavior, vol. 1, 2014.

[3] G. R. a. I. Watson, "A Review of Real-Time Strategy Game AI," 2014.

[4] Y. B. N. R. S. R. L. Vadim Bulitko, "Real-Time Heuristic Search for Pathfinding in Video Games," Artificial Intelligence for Computer Games, pp. 1-30, 2011.

[5] J. W. a. Sons, "Genetic Algorithms and Grouping Problem," Incorporated.

[6] F. Ahmed, "Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithms," Springer-Verlag Berlin Heidelberg, 2012.

[7] Y. K. C.-H. K. Shin-Jin Kang, "Live path: adaptive agent navigation in the interactive virtual world," Springer-Verlag, 2010.

[8] W. W. L. Z. Huang Jin, "Multi-Agent Pathfinding System Implemented on XNA," IEEE Computer society, 2012.

[9] [Online]. Available: http://en.wikipedia.org/wiki/Microsoft_XNA.

[10] A. K. a. I. F. I. Abdulla M. Mamdouh, "Real-time Multi-agent Simulation of Coordinated Hierarchical Movements for Military Vehicles with Formation Conservation," Proceeding of international Multi conference of engineers and computer scientists , vol. 1, 2012.

[11] J. H. Holland, "Adaptation in Natural and Artificial Systems," University of Michigan Press, Ann Arbor, 1975.

[12] K. J. Cooper T., "The Complexity of Timetable Construction Problems," Lecture Notes in Computer Science, pp. 281-295, 1996.