Cooperative Electronic Attack for Groups of Unmanned Air Vehicles based on Multi-agent Simulation and Evaluation

Yee Ming Chen[1] Wen-Yuan Wu[2]

**[1]Department of Industrial Engineering and Management, Yuan Ze University
135 Yuan-Tung Rd., Chung-Li, Tao-Yuan, Taiwan, ROC.**

[2]Department of Industrial Engineering and Management, Yuan Ze University

Abstract

In this paper, the issue of path planning is addressed for unmanned air vehicles (UAVs) cooperative joint-forces electronic attack operating in a hostile environment. Specifically, the objective is to plan path to a target location in a way that minimizes exposure to threats while keeping fuel usage at acceptable levels. We consider a scenario where a group of UAVs flies in a close formation and cooperates in their use of jamming resources to prevent being tracked by Surface-to-Air Missile (SAM) tracking radars. The main goal of this research effort is develop cooperating UAVs within multi-agent simulation environment. Simulations were generated to test the path planning and control strategies given UAVs/SAM tracking radar network scenarios, and overall UAVs cooperative electronic attack performance in each simulation was analyzed.

Keywords: *Multi-agent, Resource Allocation, Heuristically search, Path planning.*

1. Introduction

Flight path planning is the challenging technology of autonomous unmanned air vehicles (UAVs). Its intention is to work out the optimal or suboptimal safe flight trajectory in the proper time, along which UAVs are able to accomplish the prearranged task and avoid the hostile threats. Electronic Attack (EA) is a counter-counter-measure to reduce the effectiveness of detecting Radar systems to allow flight of UAVs without harm from detecting Radars and associated Surface-to-Air Missile (SAM) sites. The problem is to find a good control strategy for the UAVs carrying out the mission, including a path to follow, firing instructions and handling of unexpected events, that yields a high probability to survive[1]. To fulfill this aim, the UAVs are required to cooperate in locating and reacting to events in the environment.

The approach used is to divide the mission into a sequence of sub-problems, each forming a problem that can be solved with techniques from the fields of optimization and control [2]. The main parts in this division are timing, path planning, target assignment and target engagement. In this paper, the focus issue of flight path planning is addressed for UAVs operating in a multi-agent simulation environment. Specifically, the objective is to plan path to a target location in a way that minimizes exposure to threats while keeping fuel usage at acceptable levels. This paper presents two methods of UAV's path planning during the multi-agent circumstance, which is made up with several enemy detecting Radars. One is based on Voronoi diagram partition. We are interested in determining paths for a group of autonomous UAVs that minimize their risk of being tracked and destroyed by SAM sites. The other is the near optimal flight path can be obtained with Dijkstra heuristically search suitable for in multi-agent simulation on UAVs, which is fit for path planning offline. The planned path is only suboptimal but fit for computing and re-planning online. This research aims at explore the feasibility of using multi-agent techniques, deterministic solutions which, while suboptimal, approximate the optimal solution to within acceptable limits. The outline of this paper is as follows. UAVs flight path planning and heuristically search are presented in Section 2. Developed MASUPPE (Multi-Agent Support UAVs Path Planning Evaluation) system designs are stated in Section 3. Simulation results on evaluation of the distributed UAVs agents' simulator environment are presented in Section 4. Finally, conclusions are made in Section 5.

2. Path planning based on heuristically search

2.1 Threat network partitioning and the cost of planned

We consider a scenario in which a group of UAVs uses jamming multiple detecting Radars to avoid threats posed by SAMs and Unmanned Combat Air Vehicle (UCAV) directly destruct SAMs [3] . Figure 1. shows a hostile Integrated Air Defense Systems (IADS) multi-agent environment.

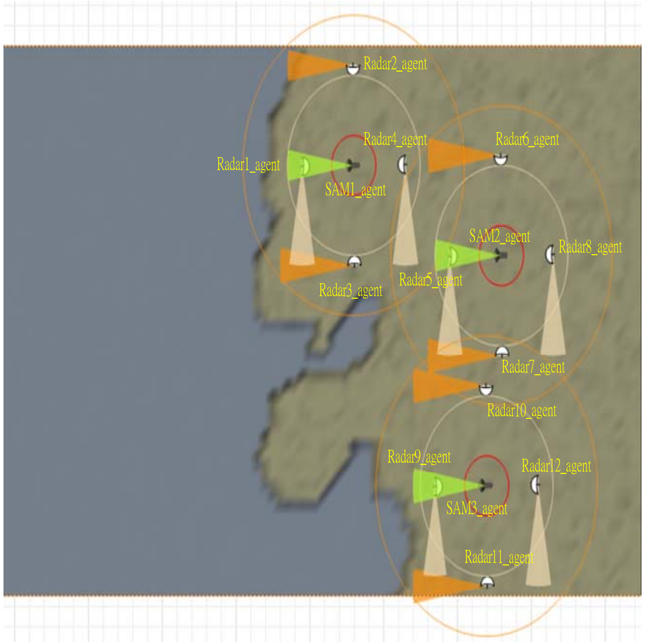

Figure 1 The multi-agent environment of IADS

A Voronoi diagram is constructed from the known IADS threat locations. Creating the Voronoi diagram entails partitioning the region of interest with threats into convex polygons or cells. Each cell contains exactly one threat, and every location within a given cell is closer to its associated threat than to any other threat. By using threat locations to create the diagram, the resulting Voronoi polygon edges form a set of lines that are equidistant from the closest threats, thereby maximizing their distance from the closest threats. Without loss of generality, the initial and target locations are not contained within cells and are connected to the nodes forming the cell [4,5]. Fig 2. shows a Voronoi diagram created for a set of threats locations.

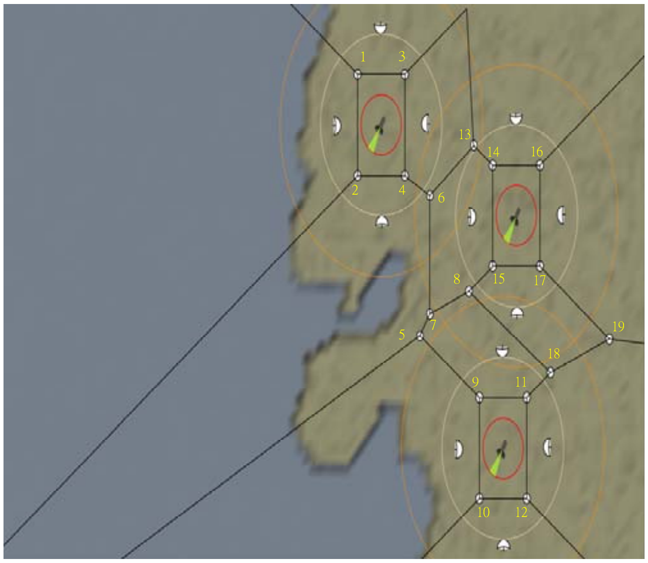

Figure 2 The Voronoi diagram partitioning of IADS

Each edge of the Voronoi diagram is assigned two costs: threat cost and length cost. Threat costs are based on a UAVs/UCAV's exposure to a radar located at the threat. Assuming that the UAVs/UCAV radar signature is uniform in all directions and is proportional to (where is the distance from the UAVs/UCAV to the threat), the threat cost for traveling along an edge is inversely proportional to the distance (to the threat) to the fourth power. In this work, the radar threaten cost was calculated at three points along each edge: $d_{1/6}$, $d_{1/2}$,and $d_{5/6}$, where d_i is the length of edge . The radar threaten cost associated with the *i*th edge is given by the expression[6,7]

$$J_{radarThreaten,i} = L_i \sum_{j=1}^{N}({}^{1}/_{d^4_{1/6i,j}} + {}^{1}/_{d^4_{1/2i,j}} + {}^{1}/_{d^4_{5/6i,j}}) \qquad (1)$$

Where N is the total number of threats (i.e. detecting Radars and SAM tracking Radars), $d^4_{1/2i,j}$ is the distance from the 1/2 point on the *i*th edge to the *j*th threat.

To include the path length as part of the cost, the length cost associated with each edge is

$$J_{lenghth,i} = L_i \qquad (2)$$

The total cost for traveling along an edge comes from a sum of the radar threaten and length costs

$$J_i = J_{radarThreaten,i} + J_{length,i} \qquad (3)$$

With the total cost determined for each of the Voronoi edges, the Voronoi diagram is searched to find the set of lowest-cost candidate paths between the initial UAVs/UCAV location and the location of the target.

2.2 Dijkstra heuristically search

Once the Voronoi diagram is complete and the UAVs/UCAV multi-agent positions and target positions are connected, a path planning method must determine the optimal path for each permutation of UAVs/UCAV multi-agent to targets. This consists of two separate parts – first, a total cost function must be developed and applied to each edge of the Voronoi diagram, and second, the edges must be searched to determine the optimal path, which is defined as the combination of graph edges that connects the UAVs/UCAV multi-agent to the targets with the lowest possible cost.

The algorithm for Dijkstra with inputs of the adjacency matrix and the beginning vertex (a UAV position) and finishing vertex (a target position) works by constructing a subgraph *S* such that the cost of any vertex *v* in *S* from the beginning vertex s is known to be minimum. The algorithm is as follows:[8,9]

```
Step 1: for each vertex $v$, set d($v$), the cost of reaching that vertex, to infinity;
Step 2. Set d(s), the cost of reaching the current vertex from itself, to zero;
Step 3. Initialize $S$ a an empty set;
Step 4. Initialize $Q$ as a set of all the vertices;
Step 5. while $Q$ still has vertices in it,
    Step a. find vertex $u$ in $Q$ that has the lowest $d(v)$ value
    Step b. include the vertex $u$ in the set $S$
        for each vertex $v$ with is connected to $u$ with an edge
            if $d(v) > d(u) + edge\ cost$
            then $d(v) = d(u) + edge\ cost$
    Step c. remove vertex $u$ from $Q$
```

This algorithm continues until the cheapest cost from the UAVs/UCAV multi-agent position to the targets positions are found.

3. The Cooperative Joint-forces EA Operation Problem

In this section, we formulate the cooperative electronic attack problem as a near optimal path planning problem which is based on the Voronoi diagram partition and Dijkstra heuristically search. A system developed is the MASUPPE (Multi-Agent Support UAVs Path Planning Evaluation). This is a distributed, real time team decision support environment comprised of simulators, entity framework, multi-agent system and user interfaces. The MASUPPE is constructed as a federation of intelligent agent-based functions that enable user-friendly scenario construction, emulation of friendly and hostile entities, and dynamic scenario control.

This research has motivated to focus on cooperative use of UAVs multi-agent for EA. Cooperative jamming is investigated from a path planning perspective. In particular, we are interested in determining paths for a group of autonomous UAVs/UCAV that minimize their risk of being tracked and destroyed by SAM sites. We, therefore, take on the flight path planning problem of joint-forces operations contain entities such as UCAV agent, UAV agents, detecting Radars agents and SAM tracking Radar agents etc. These entities for agents in a multi-agent system vary with run-time changes in situation characteristics. The UCAV agent try to penetrate/eliminate the Surface-to-Air Missile (SAM) sites, whose operations are coordinated by a command and control node. There are many situations in which the cooperation of a team of UAV agents should be paths planning to make the most efficient EA use of jamming. This is done by either distracting the Radar with confusing or deceptive information, or by blinding the Radar making it unable to detect, track, engage, or destroy threats. These works perform on minimum-risk path planning, which essentially attempted to minimize the Radar Cross Section (RCS) of the UCAV along the path [10]. Not surprisingly, jamming can have a dramatic effect in reducing risk and in general leads to drastically different minimum-risk paths [11]. According to the mission of agents in the joint-forces operations flow, we can obtain system structure of formation cooperative joint-forces combat system based on MAS, as shown in Figure. 3.

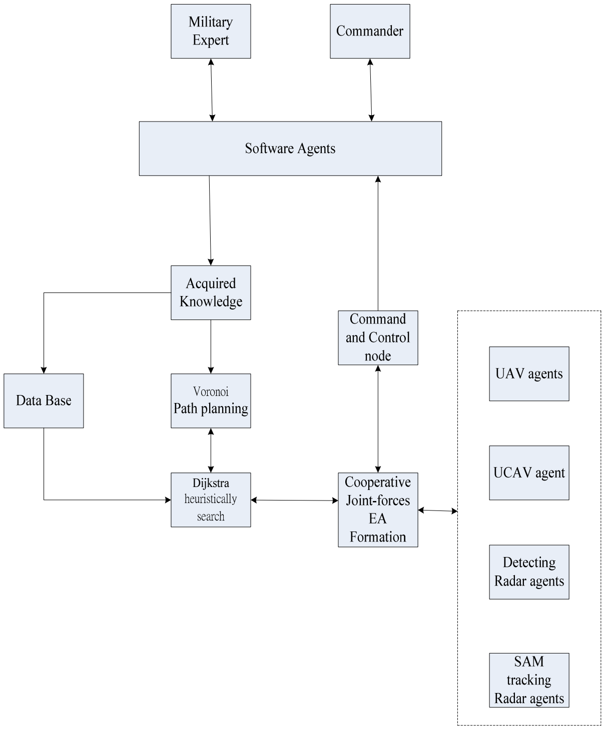

Figure 3 MASUPPE system structure of formation cooperative joint-forces EA operations

Based on above illustration, in the formation joint-forces EA operations, agents needs to establish function includes battlefield perception and interaction to command and control node which communication and exchange message with the user, moreover, as how to effectively UAVs/UCAV flight path planning and jamming targets is the main concern of the cooperative electronic attack problem.

4. Development MAS Simulation for Cooperative Joint-forces Evaluation

The main goals of the MASUPPE system are to provide: 1) a means of demonstrating realistic cooperative joint-forces EA operations scenarios, and 2) an end-to-end working system for experimentation with alternative methods for command and control. The design of the system was divided into several components: 1) the agents, 2) the agent’s environment, and 3) the behaviors for interaction among inter-agent. In designing the system, one approach would be top-down, working from developed flight path planning operations scenarios down to the agent and environment requirements. An alternative approach would be bottom-up, developing design interaction to model basic behaviors of the agents in such a way that the joint-forces EA operations could be handled indirectly [12]. Our approach is fundamentally top-down, but does have some bottom-up aspects, following something of a hybrid model.

According to the figure 3 structure that proposed above, we have constructed the MASUPPE system. Figure 4 is a screenshot of the developed MASUPPE which use java language as our program language. In order to aid the development and simulation of emergent behavior strategies for UAVs/UCAV distributed simulation is comprised of three layers: User interface, Multi-agent System (MAS) simulator, and Web based platform. The MAS simulation environment (see figure 4) is divided into three areas. The left area of Simulation Viewer is a tool to check and verify the simulation results and shows the agents’ movement and the location. The upper right two sub-windows correspond its statistics report is the simulation results. The right bottom corner is the user interface which directly provides manipulating agents’ parameters. Figure 5 shows configuration interface program provides the slide bars to directly manipulate UAV agents’ sense and detecting Radar/SAM tracking Radar search range radius for the simulation.

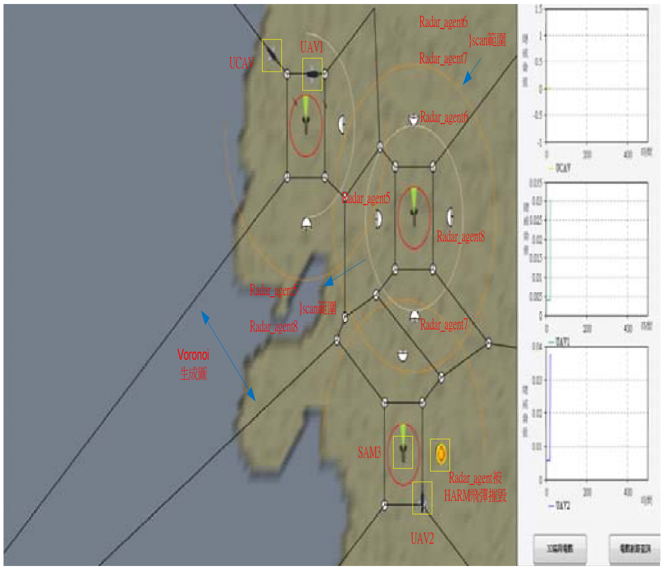

Figure 4 The MAS simulator of MASUPPE system

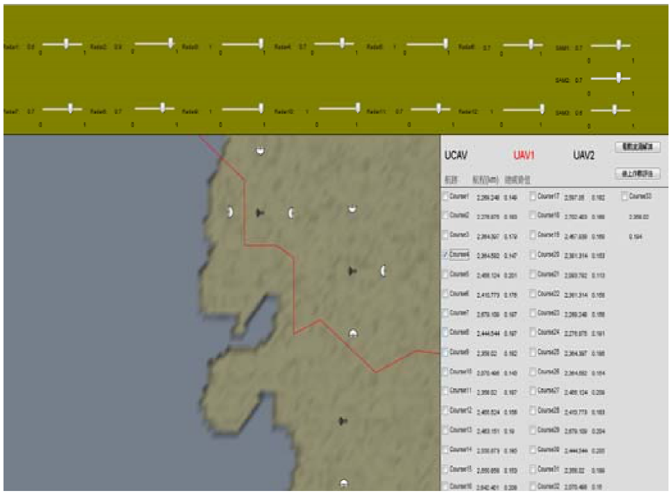

Figure 5 The User interface of MASUPPE system

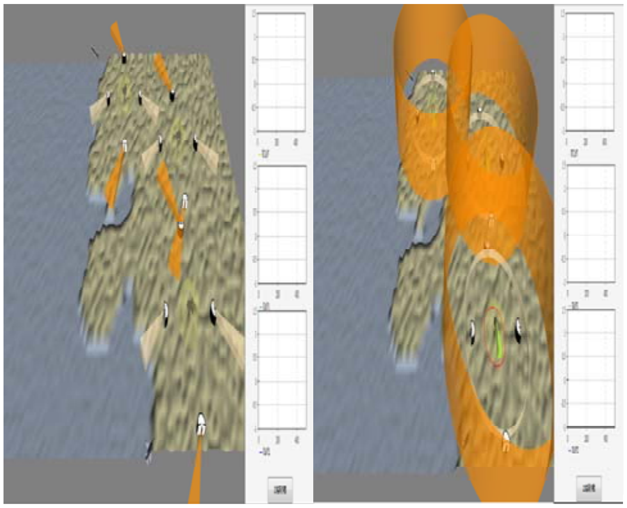

Figure 6 The detecting Radar/SAM tracking Radar agents search range

All simulation components in our MAS simulator can be classified into three categories: UAVs, UCAV, and targets (detecting Radar and SAM tracking Radar agents). UAVs/UCAV are supposed to perform their mission autonomously. With a bottom-up orientation, several low-level behavior abilities were specified. These abilities can be combined to form composite abilities. In our initial experiments, agents are executed along with the Voronoi diagram partition planning and Dijkstra heuristically search. When we perform a large number of UAV agents working together, it is necessary to test and evaluate the engineers' questions in Remote Client on the web base platform layer. The Java servlet's response is a simulation Web page with a Java applet [13,14].

MAS simulator exhibit a scheme for UAVs/UCAV working together with each other and jamming/penetrate behaviors in the 3D animation hostile environment as can be seen in Fig.6(a). A 3D animation cylinder detecting space in which detecting Radar agents search it locally and sending back messages to the SAM tracking Radar terminal end. Since the search through the three dimensional grid discussed in Fig.6(b), produced a shortest path and a cost as well as the success rate for every node in the grid (see Section 2.2 Dijkstra heuristically search), the operator can easily obtain the mission status in the case of UAVs where the goal to temporarily or permanently neutralize the IADs by forcing it to shut down or by destroying it.

MASUPPE is not only established for animation but also for on-line UAVs/UCAV joint-forces EA operation evaluation. Each agent has its own thread to progress its state. The thread execution and interaction of agents are controlled by the MAS simulator environment. With respect to UAVs and UCAV agents cooperative EA joint-forces operations, all of agents can be seen in Fig. 7(a) which highlights three UAV agents engagement. One of UAV agent jamming neutralize the detecting Radar agent. Subsequently, as observed in Fig. 7(b), which shows the consecutive snapshots of the joint-forces operations, the front SAM tracking Radar agent was eliminated. It is apparent that after some finite time, the UAVs/UCAV agents will suppress the IADs in this deployment of between detecting Radar/SAM and tracking Radar.

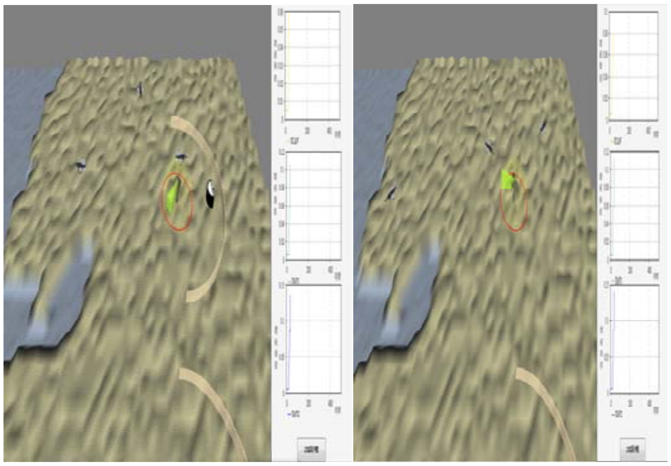

Figure 7 The consecutive snapshots of the cooperative joint-forces EA operations under UAVs/UCAV agents

5. Conclusions

In this paper, under the background of cooperative joint-forces EA operations, technique of MAS was introduced into the developed MASUPPE system. Based on the developed system, the Voronoi diagram partition planning and Dijkstra heuristically search is also presented in this hybrid model (with both top-down and bottom-up aspects). We used the Voronoi diagram partition for UAVs flight path planning and embedded Dijkstra heuristic to minimize the total cost and reduce exposure to threats environments. Future research directions include: more detailed human models; more detailed log parsing using XML; and a capability to build scenarios. The MASUPPE system also can be adapted for war game, and provides a useful set of tools for building other simulations.

Acknowledgments

This research work was sponsored by the National Science Council, R.O.C., under project number NSC101-2623-E-155-005-D.

References

[1] M. Flint, M. Polycarpou, and E. Fernandez-Gaucheand, “Cooperative control for multiple autonomous UAVs searching for targets,” *Conference on Decision and Control*, pp. 2823–2828, 2002.

[2] T. McLain, “Cooperative rendezvous of multiple unmanned air vehicles,” *AIAA Guidance, Navigation, and Control Conference*, August 2000.

[3] M. J. Mears and M. R. Akella, “Deception of radar systems using cooperatively controlled unmanned air vehicles,” *International Conference on Networking, Sensing and Control*, March 2005.

[4] A. Moitra, , R. Szczerba, , V. Didomizio, , L. Hoebel, R. Mattheyses, and B Yamrom,” A novel approach for the coordination of multi-vehicle teams. *Proc. GNC’2001*, pp. 608–618,2001.

[5] K. Tan, L. Lee, Q. Zhu, and K. Ou, “. Heuristic methods for vehicle routing problem with time windows. *Intelligent in Engineering*, pp 281–295,2006.

[6] R. M. Murray, “ Recent research in cooperative control of multi-vehicle systems”. *Journal of Dynamic Systems, Measurement and Control*, **129**(5), pp.571–583,2007.

[7] W. L. Nelson,,” Continuous curvature paths for autonomous vehicles. *IEEE Int.Conf. on Robotics and Automation*, Scottsdale, AZ, vol. 3, pp. 1260–1264,1990.

[8] A. Eele,and A. Richards,” Path planning with avoidance using nonlinear branch-and-bound optimization. *Journal of Guidance, Control, and Dynamics*, 32(2),pp.384–394,2009.

[9] S. M. LaValle, *Planning Algorithm*. Cambridge University Press, 2006.

[10] I. M. Mitcheel, and S. Sastry, “ Continuous path planning with multiple constraints”. *IEEE Conf. on Decision and Control*, pp. 5502–5507, 2003.

[11] J. L. Vian, and J. R. More,” Trajectory optimization with risk minimisation for military aircraft. *AIAA Journal of Guidance, Control and Dynamics*, **12**, pp.311–317, 1989.

[12] J. S. Shamma, , *Cooperative Control of Distributed Multi-Agent Systems*. John Wiley & Sons, Ltd, Chichester, 2007.

. [13] J. Sasiadek, and I. Duleba,” 3D local trajectory planner for UAV. *Journal of Intelligent and Robotic Systems: Theory and Applications*, **29**,pp. 191–210, 2003.

[14] Xjtechnologies software Anylogic® v6.7.1 package. http://www.xjtek.com/,2011.

Yee Ming Chen is a professor in the Department of Industrial Engineering and Management at Yuan Ze University, where he carries out basic and applied research in agent-based computing. His current research interests include soft computing, supply chain management, and system diagnosis/prognosis.

Wen-Yuan Wu received the B.S. degree in the Department of Industrial Engineering and Management at Yuan Ze University, where he has worked ever since as a research assistant, developing multi-agent system. Since 2010, he has also been studying for the Master degree at Yuan Ze University. His research interests include soft computing, multi-agent simulation, and heuristic algorithms.