

Performance Evaluation of an Unmanned Airborne Vehicle Multi-Agent System

Zhaotong Lian¹ and Abhijit Deshmukh²

¹*Faculty of Business Administration, University of Macau,*

²*Department of Industrial and Systems Engineering, Texas A&M University*

¹*China,* ²*USA*

1. Introduction

Consider an unmanned airborne vehicle (UAV) multi-agent system. A UAV agent is aware of the destination or goal to be achieved, its own quantitative or qualitative, of encountering enemy defenses in the region. Each agent plans its moves in order to maximize the chances of reaching the target before the required task completion time (see Fig. 1). The plans are developed based on the negotiations between different UAVs in the region with the overall goal in mind. The model is actually motivated by another large research project related to multi-agent systems. The information about enemy defenses can be communicated between UAVs and they can negotiate about the paths to be taken based on their resources, such as fuel, load, available time to complete the task and the information about the threat. In this system, we can also model the behavior of enemy defenses as independent agents, with known or unknown strategies. Each enemy defense site or gun has a probability of destroying a UAV in a neighborhood. The UAVs have an expectation of the location of enemy defenses, which is further refined as more information becomes available during the flight or from other UAVs. To successfully achieve the goal with a high probability, the UAVs need to select a good plan based on coordination and negotiation between each other. One paper dealing with this model is Atkins et al. (Atkins et al., 1996), which considered an agent capable of safe, fully-automated aircraft flight control from takeoff through landing. To build and execute plans that yield a high probability of successfully reaching the specified goals, the authors used state probabilities to guide a planner along highly-probable goal paths instead of low-probability states. Some probabilistic planning algorithms are also developed by the other researchers. Kushmerick et al. (Kushmerick et al., 1994) concentrate on probabilistic properties of actions that may be controlled by the agent, not external events. Events can occur over time without explicit provocation by the agent, and are generally less predictable than state changes due to actions. Atkins et al. (Atkins et al., 1996) presented a method by which local state probabilities are estimated from action delays and temporally-dependent event probabilities, then used to select highly probable goal paths and remove improbable states. The authors implemented these algorithms in the Cooperative Intelligent Real-time Control Architecture (CIRCA). CIRCA combines an AI planner, scheduler performance for controlling complex real-world systems (Musliner et al., 1995). CIRCA's planner is based on the philosophy that building a plan to handle all world

states (Schoppers, 1987)) is unrealistic due to the possibility of exponential planner execution time (Ginsberg, 1989), so it uses heuristics to limit state expansion and minimizes its set of selected actions by requiring only one goal path and guaranteeing failure avoidance along all other paths.

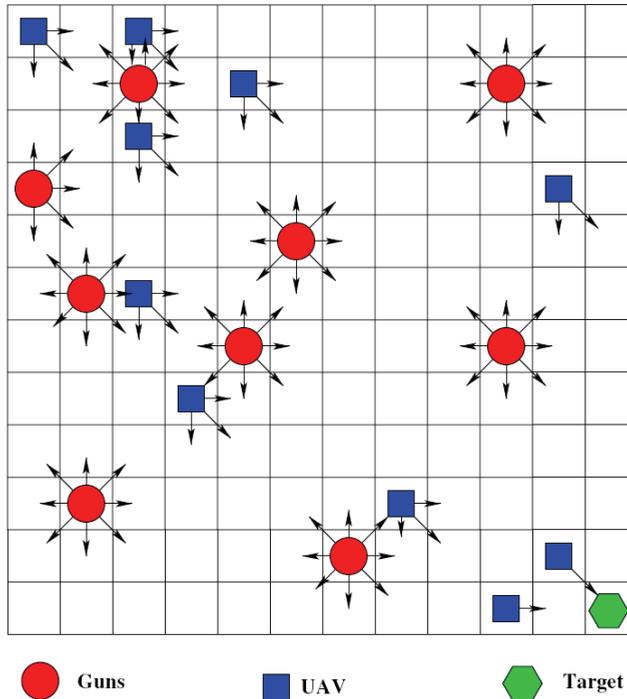


Figure 1. Unmanned aircraft system

McLain et al. (McLain et al., 2000) considered two or more UAVs, a single target in a known location, battle area divided into low threat and high threat regions by a threat boundary, and threats that 'pop up' along the threat boundary. The objective is to have the UAVs arrive at the target simultaneously, in a way that maximizes the survivability of the entire team of UAVs. The approach the authors used is to decentralize the computational solution of the optimization problem by allowing each UAV to compute its own trajectory that is optimal with respect to the needs of the team. The challenge is determine what information must be communicated among team members to give them an awareness of the situation of the other team members so that each may calculate solutions that are optimal from a team perspective.

An important methodology using in this paper is Markov decision process (MDP) based approach. An important aspect of the MDP model is that it provides the basis for algorithms that probably find optimal policies given a stochastic model of the environment and a goal. The most widely used algorithms for solving MDPs are iterative methods. One of the best known of these algorithms is due to Howard (Howard, 1960), and is known as *policy iteration*, with which, some large size MDPs (Meuleau et al., 1998; Givan et al., 1997; Littman

et al., 1995) can be solved approximately by replacing the transition probability with stationary probability.

MDP models play an important role in current AI research on planning (Dean et al., 1993; Sutton, 1990) and learning (Barto et al., 1991; Watkins & Dayan, 1992). As an extension of the MDP model, partially observable Markov decision processes (POMDP) were developed within the context of operation research (Monahan, 1982; Lovejoy, 1991; Kaelbling et al., 1998). The POMDP model provides an elegant solution to the problem of acting in partially observable domains, treating actions that affect the environment and actions that only affect the agent's state of information uniformly.

Xuan et al. (Xuan et al., 1999) considered the communication in multi-agent MDPs. Assume that each agent only observes part of the global system state. Although agents do have the ability to communicate with each other, it is usually unrealistic for the agents to communicate their local state information to all agents at all times, because communication actions are associated with a certain cost. Yet, communication is crucial for the agents to coordinate properly. Therefore, the optimal policy for each agent must balance the amount of communication such that the information is sufficient for proper coordination but the cost for communication does not outweigh the expected gain.

In this paper, we assume that there are multiple guns and UAVs in the lattice. The UAVs and guns can move to the neighboring sites at each discrete time step. To avoid the attacks from the guns, the UAVs need to figure out the optimal path to successfully reach the target with a high probability. However, a UAV cannot directly observe the local states of other UAVs, which are dynamic information. Instead, a UAV has a choice of performing a communication between two moving actions. The purpose of the communication for one UAV is to know the current local state of the other UAVs, i.e., the location and the status (dead or alive). By using the traditional MDP approach, we conduct an analytical model when there are one or two UAVs on the lattice. We extend it to a multi-UAV model by developing a heuristic algorithm.

The remainder of this paper is organized as follows. In Section 2, we derive the probability transition matrix of guns by formulate the action of guns as a Markov process. When there are only one or two UAVs in the lattice, we analyze the model as an MDP. In Section 3, we conduct extensive numerical computations. We develop an algorithm to derive the moving directions for the multi-UAV case. A sample path technique is used to calculate the probability that reaching the target is successful. Finally in Section 4, we conclude with the summary of results and suggestions for this model and the future research.

2. MDP Models

2.1 The probability transition matrix of guns

In this subsection, we discuss the action of the guns in the lattice. We assume that the size of lattice is $m_1 \times m_2$. Let $A = \{(i, j): 0 \leq i \leq m_1 - 1, 0 \leq j \leq m_2 - 1\}$ be the set of all sites in the lattice.

Each site $\mathbf{a} \in A$ is associated the number of guns $\delta_{\mathbf{a}}^t$ which can assume $q+1$ different values ($\delta_{\mathbf{a}} = 0, 1, \dots, q$) at time t . A complete set $\{\delta_{\mathbf{a}}^t, \mathbf{a} \in A, t \geq 0\}$ of lattice variables specifies a configuration of the gun system.

Since guns move to their neighbors randomly in each step without depending on their past positions, we can derive the probability transition matrix of guns by constructing a Markov

chain. When the lattice is large, however, the size of the state space becomes so big that the computation of the transition probabilities is complicated. Fortunately, the number of guns in a certain site only depends on the previous states of this site and its neighbors. we can directly derive the probability of having guns in a certain site by using some recursive formulae.

We assume that each gun has 9 possible directions to move including the current site. We denote the set of the directions by using a two-dimensional vector set $\Phi = \{(k, h) : k, h = -1, 0, 1\}$ (see Fig. 2).

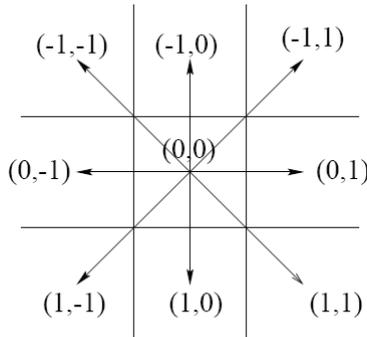


Figure 2. Walking directions of the UAVs and guns

In practice, there would not be too many guns located at one site at the same time. In order to attack UAVs more effectively, we assume that the guns negotiate with each other if there is more than one guns located in the same site. That is, they would not go to the same direction in the next step. To handle the model more easily, we restrict that there are at most 9 guns at each site.

If there is only one gun in a site, to simplify the model, we assume that the gun moves to any direction with the same probability of 1/9 including the case that the gun doesn't move at all. Obviously the probability that a gun moves to any direction is $pr(k, h, N) = N/9$ if there are N guns in the site ($N \leq 9$).

Denote $\rho^{(t)}(\mathbf{a}, n)$ as the probability that there are n guns in site \mathbf{a} at time t , where $\mathbf{a} = (a_1, a_2)$. Suppose we know $\rho^{(0)}(\mathbf{a}, n), n \leq 9$. Let's see how to calculate $\rho^{(t)}(\mathbf{a}, n)$ when $t \geq 1$ by using recursive equations.

Suppose there are j guns at site $(a_1 + k, a_2 + h)$ at time $t - 1$, then there exists one gun moving to site \mathbf{a} with probability $j/9$. Therefore, the probability that there exists one gun

$$\text{moving from site } (a_1 + k, a_2 + h) \text{ to site } (a_1, a_2) \text{ is } \sum_{j=1}^9 j \rho^{(t-1)}((a_1 + k, a_2 + h), j) / 9.$$

There will be n guns in site \mathbf{a} if there are $n_{k,h}$ guns moving from site $(a_1 + k, a_2 + h)$,

where $\sum_{k,h=-1,0,1} n_{k,h} = n$ and $n_{k,h} = 0$ or 1 . We define

$$B(n_{k,h}) = \begin{cases} \sum_{j=1}^9 j \rho^{(t-1)}((a_1+k, a_2+h), j) / 9, & n_{k,h} = 1 \\ 1 - \sum_{j=1}^9 j \rho^{(t-1)}((a_1+k, a_2+h), j) / 9, & n_{k,h} = 0 \end{cases} \quad (1)$$

It is easy to see that we have the following recursive equations:

$$\rho^{(t)}(\mathbf{a}, n) = \sum_{\substack{k,h=-1,0,1 \\ \sum n_{k,h}=n}} \prod_{k,h=-1,0,1} B(n_{k,n}) \quad (2)$$

and the stationary probability distribution $\pi(\mathbf{a}, n)$ exists:

$$\lim_{t \rightarrow \infty} \rho^{(t)}(\mathbf{a}, n) = \pi(\mathbf{a}, n) \quad (3)$$

That means, there exists a number T_0 , when $t > T_0$, $\forall \mathbf{a}$ and $n \geq 0$,

$$\rho^{(t)}(\mathbf{a}, n) \approx \rho^{(T_0)}(\mathbf{a}, n) \quad (4)$$

2.2 A general MDP model on UAVs

Since UAVs are agent-based, they determine their paths independently although they have the same global objective which is to maximize the successful probability of at least one UAV reaching the target. A UAV wouldn't know the status of other's unless they communicate with each other. We assume that there are totally N UAVs, X_0, \dots, X_{N-1} in the lattice, where N is a finite number. Let m_t be a 2^N dimensional vector standing for all communication actions of UAVs. We use 1 and 0 to represent communicating or not between two UAVs. Then m_t is a combination of 0 and 1. Let $x_i^t(m_{t-1})$ be the action of UAV X_i at time t , $i=0, \dots, N-1$. Let δ^t be the state of the guns at time t . Since the values of $(x_0^{t+1}, \dots, x_{N-1}^{t+1})$, and δ^{t+1} only depend on the values of $(x_0^t, \dots, x_{N-1}^t)$, m_t and δ^t , $\{x_0^t, \dots, x_{N-1}^t, m_t, \delta^t, t=0, 1, \dots, T\}$ consists a Markov decision process in a finite horizon T .

We can define a very simple global reward function r : any move is free and receives no reward. If one of UAVs reaches the target in t time units, the terminal reward is $\beta^t R$, where R is the probability of reaching the target and $0 \leq \beta \leq 1$ is a time discount factor. If at time T none of the UAVs reach the target, the terminal reward is 0. Each communication costs a constant c .

Denote $M^{(t)}(x_0^s(m_{s-1}), \dots, x_{N-1}^s(m_{s-1}), 1 \leq s \leq t)$ be the probability of reaching the target within t time units, $t \leq T$. Then the objective is

$$\max_t \max_{m_{s-1}, x_0^s, \dots, x_{N-1}^s, 1 \leq s \leq t} \left\{ \beta^t M^{(t)}(x_0^s, \dots, x_{N-1}^s, 1 \leq s \leq t) - c \sum_{i=0}^t m_i \mathbf{e} \right\} \tag{5}$$

where \mathbf{e} is a column vector in which all elements are 1.

Unfortunately, calculating optimal decision for (5) is not going to be computationally feasible since the combination of the decision policy is huge. Thus, we seek to reduce the size of the policy by defining approximation policies using heuristic approaches. We consider two folds. Firstly, we consider there are only two UAVs in the lattice. UAVs can communicate at every stage regardless of the history. Global states are known to both UAVs at all times, and thus we can regard it as a centralized problem where global states are observable. Once we know how to deal with the model of one or two UAVs, we can develop a scheme to handle the multi-UAV model. We will analyze it later.

Now let's consider the model with two UAVs. As a byproduct, we will see that the model with a single UAV is a special case of the model with two UAVs.

First of all, let's introduce the concept of the distance between two site $\mathbf{a} = (a_1, a_2)$ and $\mathbf{b} = (b_1, b_2)$ which is

$$S(\mathbf{a}, \mathbf{b}) = \max\{|a_1 - b_1|, |a_2 - b_2|\} \tag{6}$$

We say a vector \mathbf{b} is a neighbor of vector \mathbf{a} , if $S(\mathbf{a}, \mathbf{b}) \leq 1$. We denote the set of vector \mathbf{a} 's neighbors by $\mathcal{D}(\mathbf{a}) = \{\mathbf{b} : S(\mathbf{a}, \mathbf{b}) \leq 1\}$.

Denote $V^{(t)}(\mathbf{x}, \mathbf{y})$ as the maximum probability the UAVs successfully reach the target within t time units when both UAVs are alive and their locations are \mathbf{x}, \mathbf{y} at time 0 in the centralized sense. Denote $U^{(t)}(\mathbf{a})$ as the maximum reaching probability of a UAV within time t when there is only one UAV located at site \mathbf{a} left in the lattice. Obviously, $U^{(t)}(\mathbf{a}) = 1$ if $\mathbf{a} = \mathbf{g}$ and $V^{(t)}(\mathbf{x}, \mathbf{y}) = 1$ if $\mathbf{x} = \mathbf{g}$ or $\mathbf{y} = \mathbf{g}$, where \mathbf{g} is the position of the target location.

We can derive the rest $U^{(t)}(\mathbf{a})$, $V^{(t)}(\mathbf{x}, \mathbf{y})$ and the optimal path by the following recursive equations

$$V^{(0)}(\mathbf{x}, \mathbf{y}) = \begin{cases} 1, & \mathbf{x} = \mathbf{g}, \text{ or } \mathbf{y} = \mathbf{g}, \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

$$U^{(0)}(\mathbf{a}) = \begin{cases} 1, & \mathbf{a} = \mathbf{g}, \\ 0, & \mathbf{a} \neq \mathbf{g}, \end{cases} \tag{8}$$

$$U^{(t)}(\mathbf{a}) = \max_{\bar{\mathbf{a}} \in \mathcal{D}(\mathbf{a})} \left\{ \rho^{(T-t)}(\bar{\mathbf{a}}, \mathbf{0}) U^{(t-1)}(\bar{\mathbf{a}}) \right\} \tag{9}$$

and

$$V^{(t)}(\mathbf{x}, \mathbf{y}) = \max_{\bar{\mathbf{x}} \in \mathcal{D}(\mathbf{x}), \bar{\mathbf{y}} \in \mathcal{D}(\mathbf{y})} \left\{ \begin{aligned} & \rho^{(T-t)}(\bar{\mathbf{x}}, \mathbf{0}) V^{(t-1)}(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \\ & + \rho^{(T-t)}(\bar{\mathbf{x}}, \mathbf{0}) (1 - \rho^{(T-t)}(\bar{\mathbf{y}}, \mathbf{0})) U^{(t-1)}(\bar{\mathbf{x}}) \\ & + \rho^{(T-t)}(\bar{\mathbf{y}}, \mathbf{0}) (1 - \rho^{(T-t)}(\bar{\mathbf{x}}, \mathbf{0})) U^{(t-1)}(\bar{\mathbf{y}}) \end{aligned} \right\}, \quad 1 \leq t \leq T \tag{10}$$

where T is the lifetime of the UAV at time 0.

We denote the algorithm based on the above formula as **Double-MDP**. Usually, we have to use the classical dynamic programming to solve the above MDP problem. Note that only the first step is optimal because we assume that the UAVs communicate all the time. Once finish the first step, they have to figure out the status of each other (dead or alive) again. Obviously, the game is over if both UAVs die. If both are still alive, we can repeat the above MDP to obtain the optimal paths. If only one UAV is alive, we will use (9) to obtain the optimal path for this UAV.

Specially, when there is only one UAV in the lattice at the beginning, we can calculate the optimal path and the maximum successful probability only by using (8) and (9). We call the algorithm based on one UAV a **Single-MDP**. Even there are more than one UAV in the lattice, we still call it Single-MDP based approach if only they find their moving directions independently according to their own local objectives.

Intuitively, the UAVs affect each other only when they are close, for instance, when they are neighbors. Hence we can develop an heuristic algorithm in which, UAVs communicate with each other only when they are neighbors. Since the agent based UAVs know each other at the beginning, and they use the same Double-MDP approaches, they should know when they are neighbors at time 0 or at the time of their last communication.

Furthermore, we can improve the above algorithm by extend the definition of neighbor for UAVs, in which, the UAVs communicate with each other only when they are neighbors.

Definition 1. We say two UAVs located in \mathbf{a} and \mathbf{b} are neighbors if their distance $S(\mathbf{a}, \mathbf{b}) \leq d$, where d is a non-negative integer.

When $d = 0$, the UAVs communicate when they are in the same site;

When $d = 1$, the UAVs communicate when they are in the same site or they are 'real' neighbors;

When $d > m$ where $m \times m$ is the size of the lattice, the UAVs communicate with each other in all steps.

To evaluate the above algorithm and choose a suitable d , we have to calculate successful probabilities for each d . Unfortunately, it is hard to obtain the successful probabilities analytically. What we are going to do is to calculate the successful times by combining the MDP and sample path technique. We generate the gun's status (guns exist or not) on the location of the UAVs according to the transition probability of guns. If there exist guns in the location of a UAV, the UAV is killed. If both of UAVs die before the reach the target, we say the UAVs fail, otherwise, we say the UAVs success.

2.3 Negotiation between UAVs

Besides Double-MDP approach, we can also consider that two UAVs derive their moving directions by using Single-MDP. They may negotiate with each other when they are neighbors by changing directions. The basic idea is let the UAVs negotiate when they have the same optimal direction in the next step. We call this Single-MDP based approach **Nego-MDP**. Since Nego-MDP is also a one-dimensional MDP, it should be much faster than Double-MDP to obtain the numerical results. Assume that both UAVs A and B have the same first choice \mathbf{c} at time t . The successful probabilities are $A.p(\mathbf{c})$ and $B.p(\mathbf{c})$ respectively. A and B have the second choices \mathbf{a} and \mathbf{b} (see Figure 3). The successful

probabilities are $A.p(\mathbf{a})$ and $B.p(\mathbf{b})$ respectively. Let $\rho^{(t+1)}(\mathbf{x}, 0)$ be the probability of having no gun in the position \mathbf{x} at time $t + 1$. We define the following probabilities.

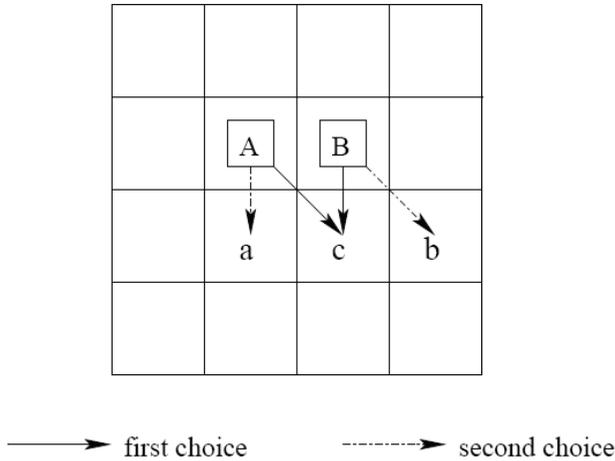


Figure 3. Negotiation Analysis

P_1 : the successful probability of both A and B going to site \mathbf{c} at the time $t + 1$.

P_2 : the successful probability of A going to site \mathbf{c} and B going to site \mathbf{b} at time $t + 1$.

P_3 : the successful probability of A going to site \mathbf{a} and B going to site \mathbf{c} at time $t + 1$.

We have

$$P_1 = \left(V_A^{(T_A-t)}(\mathbf{c}) + V_B^{(T_B-t)}(\mathbf{c}) - V_A^{(T_A-t)}(\mathbf{c})V_B^{(T_B-t)}(\mathbf{c}) \right) p^{(t+1)}(\mathbf{c}, 0) \tag{11}$$

$$P_2 = V_A^{(T_A-t-1)}(\mathbf{c}) + V_B^{(T_B-t-1)}(\mathbf{b}) - V_A^{(T_A-t-1)}(\mathbf{c})V_B^{(T_B-t-1)}(\mathbf{b}) \tag{12}$$

$$P_3 = V_A^{(T_A-t-1)}(\mathbf{a}) + V_B^{(T_B-t-1)}(\mathbf{c}) - V_A^{(T_A-t-1)}(\mathbf{a})V_B^{(T_B-t-1)}(\mathbf{c}) \tag{13}$$

where $V_A^{(T_A-t)}(\mathbf{a})$ stands for the successful probability for aircraft A starting from site \mathbf{a} at time t when the initial gas is T_A .

Comparing among these three probabilities, we choose the corresponding activity when the successful probability is maximum (see Fig. 4). That is,

Case 1: P_1 is the maximum, both A and B go to site \mathbf{c} .

Case 2: P_2 is the maximum, A goes to \mathbf{c} , B goes to site \mathbf{b} .

Case 3: P_3 is the maximum, A goes to \mathbf{a} and B goes to site \mathbf{c} .

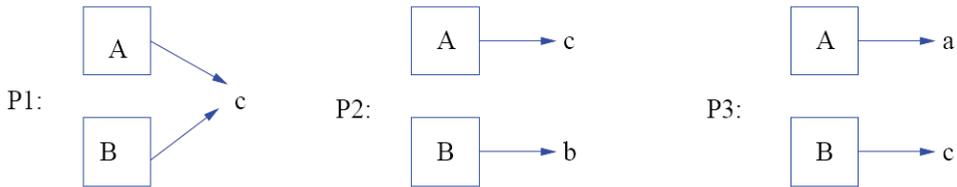


Figure 4. Negotiation Analysis

Once we have the results of single UAV and double UAV models, we can apply the Single-MDP, Nego-MDP and Double-MDP to the multi-UAV model numerically. We will discuss that in detail in the next section.

3. Numerical Analysis

In the section, we analyze the UAV model numerically. We firstly discuss the communication issue in the case with two UAVs. Then we compare the successful probabilities among the Single-MDP, Nego-MDP and Double-MDP approach in the multi-UAV model.

In the following examples, we assume that the size of the lattice is $m \times m$, where $m = 15$, and the target is located at $\mathbf{g} = (m-1, m-1)$. The amount of the gas in each UAV is 20 units. We assume each unit time the UAV needs to spend a unit gas. The probability $pr(k, h) = 1/9, \forall k, h = -1, 0, 1$.

A C++ program is written to calculate the successful probabilities and the optimal path. First of all, we calculate $\rho^{(t)}(\mathbf{a}, 0)$ and save them as an array $p0[x][y][t]$. We then calculate $U^{(t)}(\mathbf{a})$ and $V^{(t)}(\mathbf{x}, \mathbf{y})$ by using (7), (8), (9) and (10), and save them as arrays $U[x][y][t]$ and $V[x_1][y_1][x_2][y_2][t]$ respectively.

3.1 Double-MDP in the two-UAV model

In this subsection, we assume that there are only two UAVs in the lattice. By using Double-MDP algorithm, we are going to detect how the successful probabilities are different between communication or non-communication.

The UAVs start from the sites $(0, posi)$ and $(posi, 0)$, where $posi \leq m-1$. Since we assume that the UAVs know each other at the beginning, they can calculate the whole moving path for both UAVs individually because they are using the same algorithm. When they are neighbors, they will communicate with each other to see the status of the other UAV (dead or alive). We generate the number of guns at a certain site according to the probability of having guns at this site to determine whether the UAVs are dead or not. The game is over when all UAVs die or at least one UAV reach the target. By using the sample path technique, we can simulate the whole procedure. Below is the pseudo-code to describe how the UAVs find the path to reach the target.

```

-----
gas=20; t=0; \\initializing
While(at least one UAV is alive){
  if(both UAVs are alive){
    Obtain the optimal directions for both UAVs according to
    the Double-MDP and move one step;
    Generate the status of guns on the locations of both UAVs;
  }
  else{
    Obtain the optimal direction for the alive UAV according
    to the Single-MDP and move one step;
    Generate the status of guns on the location of the UAV;
  }
  if(At least one UAV reaches the target){
    break;
  }
  else{
    gas++; t++;
  }
}
-----

```

gun rate = 0.09									
d \ posi	0	1	2	3	4	5	6	7	8
0	0.568	0.543	0.543	0.357	0.370	0.397	0.427	0.455	0.469
1	0.586	0.562	0.562	0.525	0.370	0.397	0.427	0.455	0.469
2	0.586	0.562	0.562	0.529	0.370	0.397	0.427	0.455	0.469
3	0.586	0.562	0.562	0.529	0.370	0.397	0.427	0.455	0.469
4	0.586	0.562	0.562	0.529	0.370	0.397	0.427	0.455	0.469
gun rate = 0.12									
0	0.481	0.455	0.455	0.427	0.227	0.215	0.218	0.233	0.257
1	0.500	0.476	0.476	0.443	0.227	0.215	0.218	0.233	0.257
2	0.500	0.476	0.476	0.447	0.227	0.215	0.218	0.233	0.257
3	0.500	0.476	0.476	0.447	0.227	0.215	0.218	0.233	0.257
4	0.500	0.476	0.476	0.447	0.237	0.218	0.218	0.234	0.256
gun rate = 0.15									
0	0.425	0.400	0.400	0.360	0.176	0.164	0.169	0.181	0.200
1	0.446	0.421	0.421	0.375	0.176	0.164	0.169	0.181	0.200
2	0.446	0.421	0.421	0.379	0.176	0.164	0.169	0.181	0.200
3	0.446	0.421	0.421	0.379	0.176	0.164	0.170	0.182	0.200
4	0.446	0.421	0.421	0.379	0.176	0.165	0.171	0.182	0.201

Table 1. Successful probabilities for symmetric UAVs

We consider two different cases for guns at time t : symmetric and asymmetric cases. In the symmetric case, we assume $p_0[x][y][0]=0.09, 0.12, \text{ or } 0.15$ for all (x, y) where $p_0[x][y][0]$ is the probability that there is no gun at site (x, y) at time 0. We call these probabilities **gun rates**. In the asymmetric case, we assume $\text{gun rate}=0.03, 0.06, \text{ or } 0.09$ only when $x > y$. $p_0[x][y][0]=0.0$ for the other (x, y) . We consider the neighbor meter $d=0, 1, 2, 3 \text{ or } 4$. At time 0, the UAVs are located at $(0, \text{posi})$ and $(\text{posi}, 0)$, where $\text{posi}=0, 1, \dots, 8$. Each simulation, we take 30 different seeds, and run 500 replications for each seed, then we calculate the averages and the coefficients of the successful probability. We found that the coefficients are about 5% which is acceptable. Table 1 and 2 are the numerical results.

From the Table 1 and Table 2, we see that the maximum differences of successful probabilities between non-communication ($d=0$) and communication ($d>0$) are not significantly different. And only when both UAVs start from the same site $(0, 0)$, the differences reach 2%. The conclusion is that it is not necessary to communicate with each other to know whether another UAV is still alive or not. Specially when they are not neighbors.

gun rate = 0.03									
$d \setminus \text{posi}$	0	1	2	3	4	5	6	7	8
0	0.984	0.984	0.985	0.985	0.985	0.985	0.985	0.978	0.982
1	0.984	0.985	0.985	0.985	0.985	0.985	0.985	0.978	0.982
2	0.985	0.985	0.986	0.986	0.986	0.986	0.986	0.978	0.983
3	0.985	0.986	0.986	0.986	0.986	0.986	0.986	0.980	0.984
4	0.985	0.986	0.986	0.986	0.986	0.986	0.986	0.980	0.984
gun rate = 0.06									
$d \setminus \text{posi}$	0	1	2	3	4	5	6	7	8
0	0.896	0.914	0.919	0.924	0.921	0.913	0.913	0.900	0.929
1	0.897	0.916	0.919	0.924	0.921	0.913	0.913	0.900	0.929
2	0.897	0.915	0.921	0.926	0.921	0.913	0.913	0.900	0.931
3	0.896	0.916	0.921	0.927	0.924	0.916	0.916	0.900	0.932
4	0.896	0.916	0.921	0.927	0.925	0.918	0.918	0.904	0.932
gun rate = 0.09									
$d \setminus \text{posi}$	0	1	2	3	4	5	6	7	8
0	0.795	0.808	0.823	0.823	0.806	0.786	0.786	0.786	0.810
1	0.797	0.811	0.823	0.823	0.806	0.786	0.786	0.786	0.810
2	0.798	0.812	0.826	0.823	0.806	0.786	0.786	0.786	0.810
3	0.797	0.813	0.827	0.830	0.813	0.786	0.786	0.786	0.810
4	0.797	0.813	0.827	0.830	0.814	0.796	0.796	0.796	0.817

Table 2. Successful probabilities for symmetric UAVs

3.2 Multi-UAV

Based on the conclusion of the case with two UAVs, we can develop a scheme to deal with the case of more than two UAVs. We know that the UAVs affect each other only when they get close. As an intuitive algorithm, we let the UAVs group themselves two by two dynamically based on their distance at the beginning.

In the following examples, we consider the symmetric case. We assume $gun\ rate = 0.02, 0.04, \dots, 0.18$. We also symmetrically launch the UAVs from the sites $(0, posi)$ and $(posi, 0)$, $0 \leq posi \leq 14$ depending on the number of UAVs. There is at most only one UAV each site. For example, if there is only one UAV, we launch it from $(0, 0)$; if there are 5 UAVs, we launch them from $(2, 0)$, $(1, 0)$, $(0, 0)$, $(0, 1)$ and $(0, 2)$ etc. We assume that the number of UAVs are $1, 3, 5, \dots, 2m-1$ so that the UAVs are launched symmetrically. For example, if we have 5 UAVs, we group them as three groups: $\{(2, 0), (1, 0)\}$, $\{(0, 0), (0, 1)\}$ and $\{(0, 2)\}$. Since all UAVs use the same algorithm in each experiment and they know each other at the beginning, they can figure out the moving directions of all UAVs individually without communication. When one or more UAVs reach the target, we say the UAVs success. Below is the algorithm based on the Double-MDP. The algorithm based on the Nego-MDP is similar except that we need to replace Double-MDP with Nego-MDP in step III.

- I. Let $gas = 20$ and $t = 0$;
- II. Group UAVs two by two;
- III. Obtain the optimal paths for the alive UAV according to the Single-MDP or Double-MDP and move one step;
- IV. Generate the status of guns on the locations of UAVs and see if the UAVs will be attacked by the guns;
- V. If at least one UAV reaches the target, the UAV is successful and stop; otherwise, let $gas = gas - 1$, $t = t + 1$ and go back to III;

Comparing with the Double-MDP or Nego-MDP algorithm, the Single-MDP algorithm is simpler, in which, all UAVs figure out their moving directions independently based on the Single-MDP algorithm till one of UAVs reaches the target.

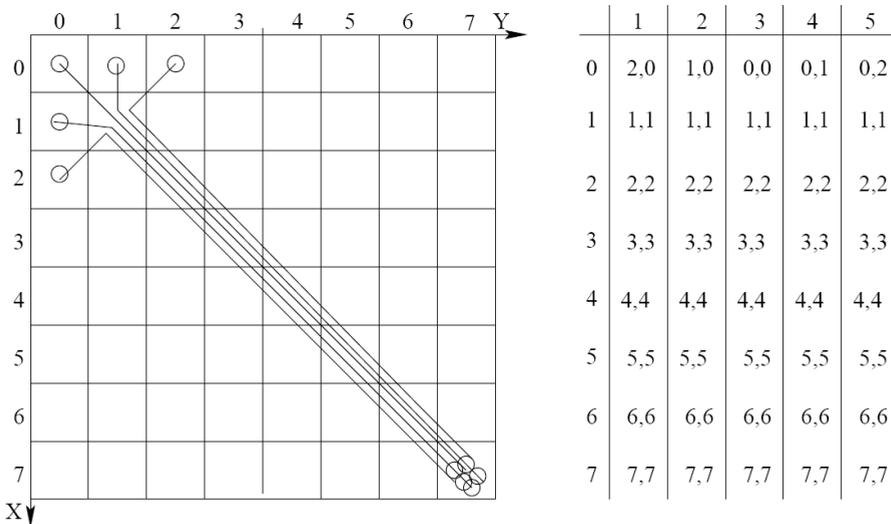


Figure 5. A sample of moving paths for Single-MDP

Fig. 5, 6 and 7 are the sample paths based on the Single-MDP, Nego-MDP and Double-MDP respectively when the size of lattice is 8×8 and the gas of the UAV is 15 units. We assume

there are 5 UAVs and $gun\ rate = 0.10$. In Fig. 5, since the UAVs independently make decision based on their own objective, they have the same local optimal path: (1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6), (7, 7). In Fig. 6 and 7, UAVs cooperate two by two. UAV 1 and UAV 2 have the different moving paths, so do UAV 3 and UAV4, but UAV 5 still acts independently. On the other hand, Group 1 (UAV 1, 2) and Group 2 (UAV 3, 4) are independent, they have the similar paths. Obviously, if we increase the group size, or let all UAVs cooperate with each other, the successful probability can still be improved. But it will increase the complexity of the algorithm.

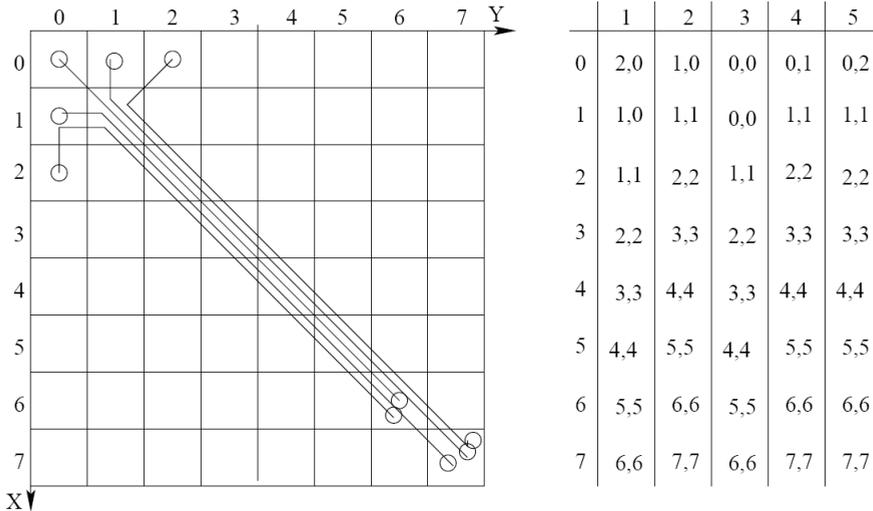


Figure 6. A sample of moving paths for Nego-MDP

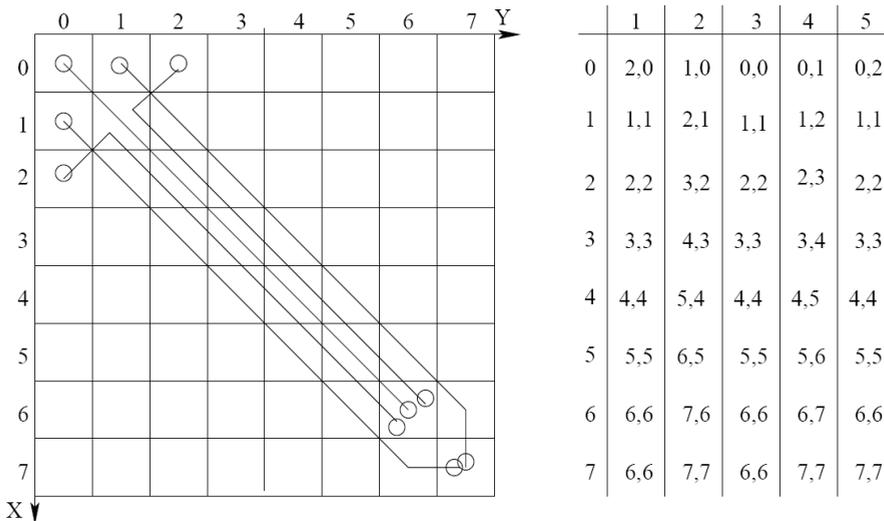
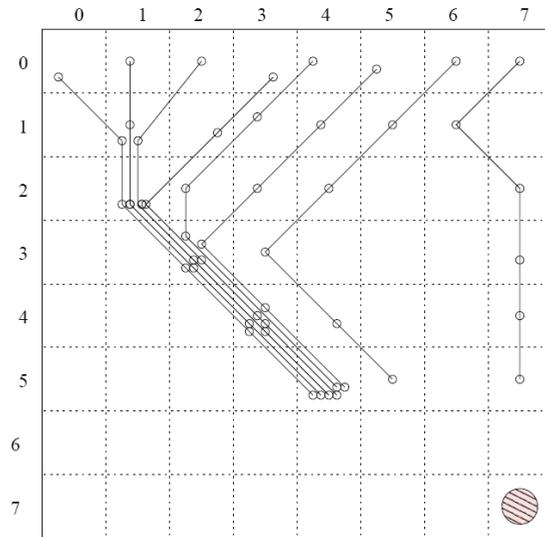


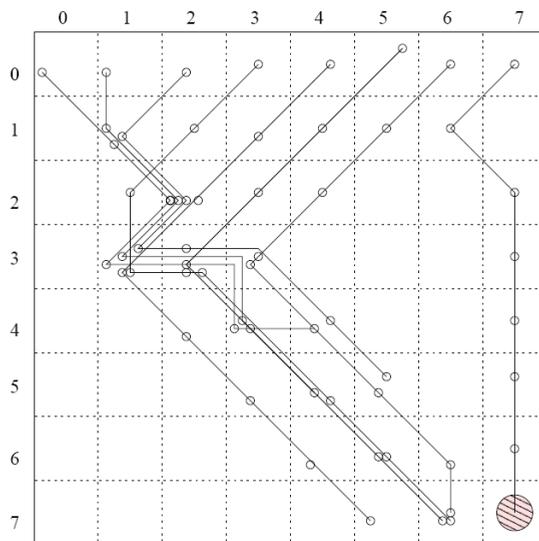
Figure 7. A sample of moving paths for Double-MDP

Fig. 8 and 9 also show the sample paths based on the Single-MDP and Nego-MDP for the asymmetric case when the gun rate=0.65 in a site (i,j) if $i < j$, and the gun rate= 0.55 if $i \geq j$. We can see the paths are very different between the Single-MDP and Nego-MDP.



Sample paths for independent case

Figure 8. Sample paths for Single-MDP (asymmetric guns)



Sample paths for negotiating case

Figure 9. Sample paths for Nego-MDP (asymmetric guns)

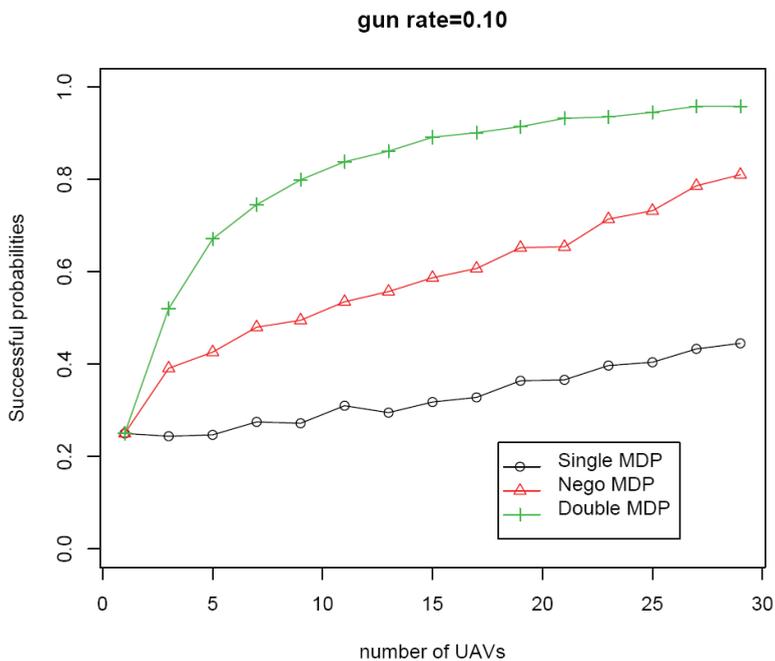


Figure 10. Successful probabilities for different numbers of UAVs

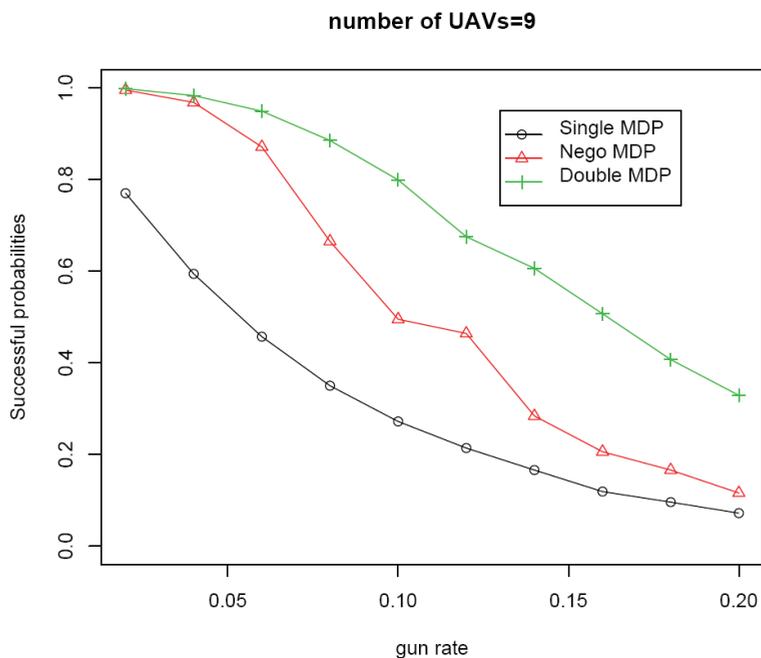


Figure 11. Probabilities of successfully reaching the target for different gun rates

Fig. 10 shows the relation between the successful probability and the number of UAVs. Fig. 11 shows the relation between the gun rate and the probability of successfully reaching the target. Obviously, for all three algorithms, the probabilities of reaching the target are increasing with the number of UAVs, and decreasing with the gun rate. We found that the Double-MDP algorithm is the best algorithm. Nego-MDP is much better than the Single-MDP algorithm. The average probability difference between Double-MDP and Nego-MDP are about 0.2. From Fig. 11, however, we see both Double-MDP and Nego-MDP reach 0.95 when the gun rate is less than 0.05, while the Single-MDP only reach about 0.7. Since Nego-MDP is a one-dimensional MDP based on the Single-MDP algorithm, it is much faster than Double-MDP which is two-dimension MDP (see Fig. 12 and 13). When the gun rate is smaller, Nego-MDP is good enough to use. When the gun rate is larger, however, the successful probability of Nego-MDP is close to the successful probability of Single-MDP. In this case, we recommend to use double MDP. From Fig. 11, we can see that the successful probability for the Double-MDP is increasing concave function of the UAV launching rate, which means that the successful probabilities would not increase significantly when the launching rate is large. This result tells us that it is not necessary to launch so many UAVs in order to reach the target with a certain probability.

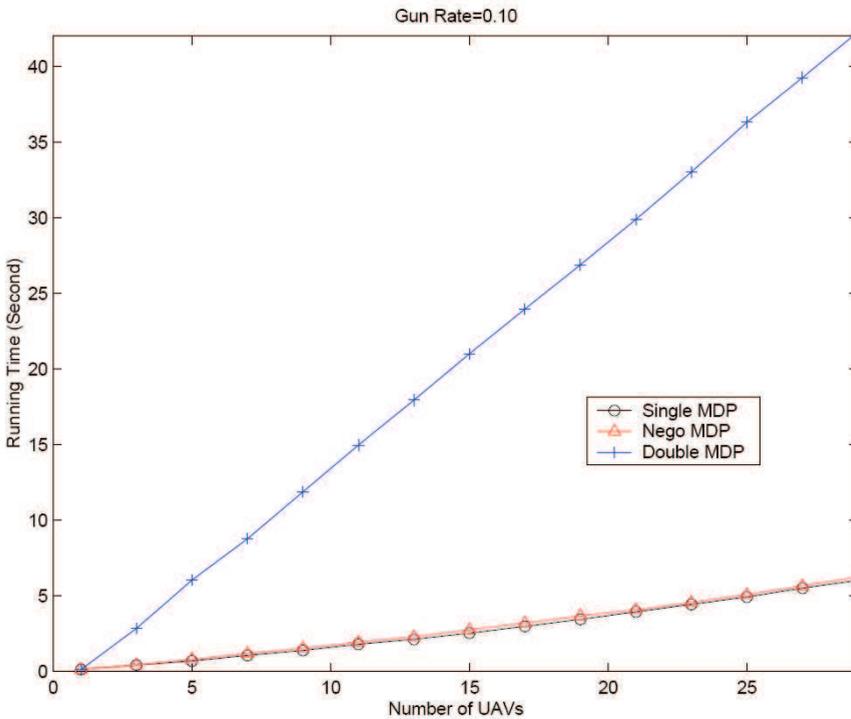


Figure 12. Running time comparison among Single-MDP, Nego-MDP and Double-MDP

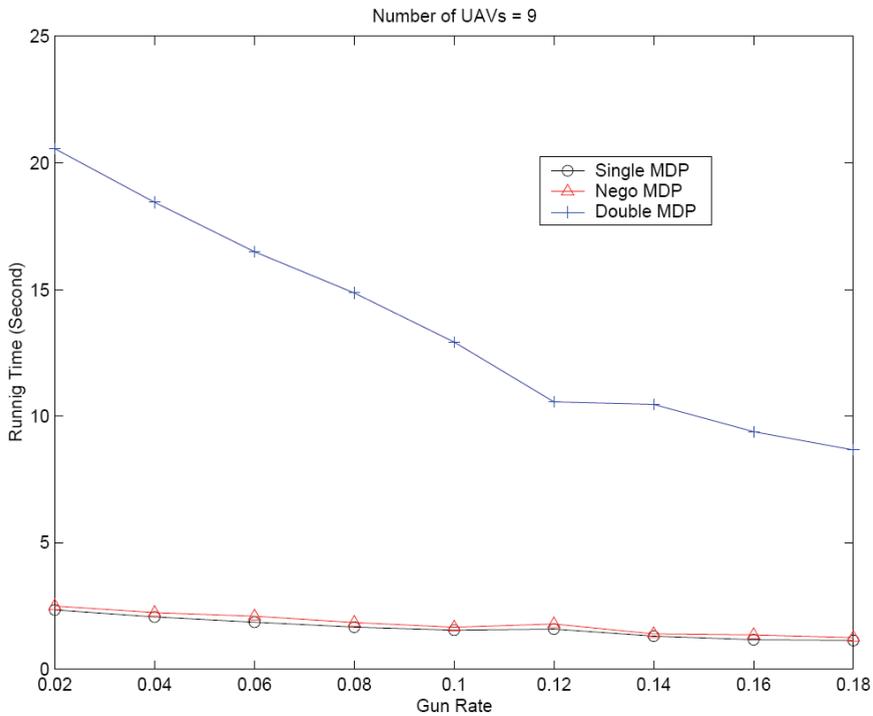


Figure 13. Running time comparison among Single-MDP, Nego-MDP and Double-MDP

4. Summary and Future Work

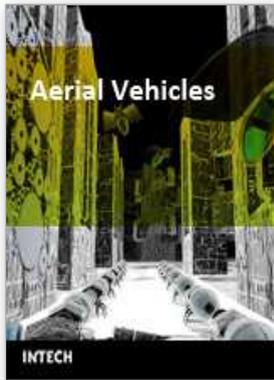
In this paper, we study a multi-agent based UAV system. Centralized MDP is complicated and unrealistic. In this decentralized model, UAVs have the same global objective which is to reach the target with maximum probability, but each UAV can make decision individually. Although the optimality problem is computational prohibitive, the heuristic results give us very important managerial insights. Based on the Nego-MDP and the Double-MDP algorithms, UAVs group themselves two by two dynamically, and find out the moving directions very effectively. Obviously, increasing the group size can improve the successful probability, but in the mean time, the algorithm will become more complicated. The precise information on guns are very important for UAVs to reach the target effectively. So far, we only consider that the guns randomly walk on the lattice. That will be worthwhile if we can update the gun information dynamically. Further more, it will be interesting if UAVs can also attack guns. In that case, we need to introduce the game theory to figure out the best strategy for both the guns and the UAVs.

5. Acknowledgements

This work was funded in part by NSF Grants # 0075462, 0122173, 0325168, AFRL Contract # F30602-99-2-0525, and by UMAC Grant # RG016/02-03S.

6. References

- Atkins, E.M.; Durfee, E.H. & Shin, K.G. (1996). Plan development using local probabilistic models, *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pp. 49-56
- Barto, A.G.; Bradtke, S.J. & Singh, S.P. (1991). *Real-time learning and control using asynchronous dynamic programming*, Technical Report 91-57, University of Massachusetts, Amherst, Massachusetts
- Dean, T.; Kaelbling, L.P.; Kirman, J. & Nicholson, A. (1993). Planning with deadlines in stochastic domains, *Proceedings of AAAI*, pp.574-579
- Ginsberg, M.L. (1989). Universal planning: An (almost) universally bad idea. *AI Magazine*, Vol. 10, No. 4
- Givan, R.; Leach, S.M. & Dean, T. (1997). *Bounded parameter Markov decision processes*, Vol. 1348, pp. 234-246, Springer
- Howard, R.A. (1960). a. Dynamic programming and Markov processes, b. The Technology Press of The Massachusetts Institute of Technology and John Wiley & Sons, Inc.
- Kaelbling, L.P.; Littman, M.L. & Cassandra, A.R. (1998). Planning and acting in partially observable stochastic domains, *Artificial Intelligence* 101
- Kushmerick, N.; Hanks, S. & Weld, D. (1994). An algorithm for probabilistic least-commitment planning, *Proceeding of AAAI*, pp. 1073-1078
- Littman, M.L.; Dean, T.L. & Kaelbling, L.P. (1995). On the complexity of solving Markov decision problems, *Proceedings of UAI-95*
- Lovejoy, W.S. (1991). A survey of algorithmic methods for partially observed Markov decision processes, *Annals of Operations Research*, Vol. 28, pp. 47-65
- McLain, T.W.; Chandler, P.R. & Pachter, M. (2000). A decomposition strategy for optimal coordination of unmanned air vehicles, *Proceedings of the American Control Conference*, pp. 369-373, Chicago
- Meuleau, N.; Hauskrecht, M.; Kim, K.E.; Peshkin, L.; Kaelbling, L.P.; Dean, T. & Boutillier G. (1998). Solving very large weakly coupled Markov decision processes, *Proceedings of the Fifteenth National Conference on Artificial Intelligence*
- Monahan, G.E. (1982). A survey of partially observable Markov decision processes: Theory, models, and algorithms, *Management Science*, Vol. 28, pp. 1-16
- Musliner, D.J.; Durfee, E.H. & Shin, K.G. (1995). World modeling for the dynamic construction of real-time control plans, *Artificial Intelligence*, Vol. 74, pp. 83-127
- Schoppers, M.J. (1987). Universal plans for reactive robots in unpredictable environments, *Proceeding of International Joint Conference on Artificial Intelligence*, pp. 1039-1046
- Sutton, R.S. (1990). Integrated architectures for learning, planning and reacting based on approximating dynamic programming, *Proceedings of the seventh International Conference on Machine Learning (Austin, Texas)*, Morgan Kaufmann
- Watkins, C.J. & Dayan, P. (1992). Q-learning, *Machine Learning*, Vol. 8, pp. 279-292
- Xuan, P.; Lesser, V. & Zilberstein, S. (1999). Communication in multi-agent Markov decision processes, *Technical report*, University of Massachusetts at Amherst



Aerial Vehicles

Edited by Thanh Mung Lam

ISBN 978-953-7619-41-1

Hard cover, 320 pages

Publisher InTech

Published online 01, January, 2009

Published in print edition January, 2009

This book contains 35 chapters written by experts in developing techniques for making aerial vehicles more intelligent, more reliable, more flexible in use, and safer in operation. It will also serve as an inspiration for further improvement of the design and application of aerial vehicles. The advanced techniques and research described here may also be applicable to other high-tech areas such as robotics, avionics, vetronics, and space.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Zhaotong Lian and Abhijit Deshmukh (2009). Performance Evaluation of an Unmanned Airborne Vehicle Multi-Agent System, Aerial Vehicles, Thanh Mung Lam (Ed.), ISBN: 978-953-7619-41-1, InTech, Available from: http://www.intechopen.com/books/aerial_vehicles/performance_evaluation_of_an_unmanned_airborne_vehicle_multi-agent_system

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.