11th IFAC International Workshop on
Adaptation and Learning in Control and Signal Processing
July 3-5, 2013. Caen, France

Optimization of Decentralized Task Assignment for Heterogeneous UAVs ⋆

Dong Jun Kwak * **Sungwon Moon** * **Suseong Kim** *
H. Jin Kim *

* *Department of Mechanical and Aerospace Engineering
Seoul National University, Seoul, Korea
(e-mail: djunkwak@gmail.com, sungwon.moon1@gmail.com,
suseongkim@snu.ac.kr, hjinkim@snu.ac.kr)*

Abstract: In this paper, the optimization of a decentralized task assignment strategy for heterogeneous UAVs in a probabilistic engagement scenario is investigated. In the engagement scenario, each UAV selects its targets by employing the consensus-based bundle algorithm (CBBA). This paper uses a scoring matrix to reflect heterogeneity among the UAVs and targets with different capabilities. Therefore, a performance improvement of CBBA is closely connected with the scoring matrix and it should be optimally selected. The values of scoring matrix can be obtained by employing an episodic parameter optimization (EPO). The EPO algorithm is performed during the numerous repeated simulation runs of the engagement and the reward of each episode is updated using reinforcement learning. The candidate scoring matrices are selected by using particle swarm optimization. The optimization results show that the team survivability of the UAVs is increased after performing the EPO algorithm and the values of the optimized score matrix are also optimally selected.

Keywords: Multi-agent system, Heterogeneous UAVs, Decentralized task assignment, CBBA.

1. INTRODUCTION

Recent advances in intelligent robotic systems have allowed exploitation of autonomous vehicles in various fields, and many researchers have studied multi-agent problems such as coordination, mission assignment, search optimization, rendezvous, etc (Beard et al. (2002), Jin et al. (2003), Chung and Burdick (2012), Yao et al. (2010)). When heterogeneous vehicles are employed to perform complicated tasks that are impossible for a single vehicle, these tasks should be appropriately assigned to individual vehicles by considering the characteristics of each vehicle. This paper deals with the task assignment problem for heterogeneous unmanned aerial vehicles (UAVs) in a probabilistic engagement scenario against hostile elements.

Centralized methods are the traditionally used to solve the task assignment problem (Bellingham et al. (2001)). In this approach, a specific UAV performs the role as a leader and plans its missions as well as the others. Then, the leader UAV sends the information of the planned tasks to others through communication channels. This approach results in a simpler communication structure as the UAVs do not have to communicate among themselves, but a heavy computational load is placed on the leader UAV. Moreover, in this structure the loss or malfunction of the leader UAV can result in a total breakdown of the system. The other approach is the decentralized task assignment in which each UAV makes decisions for itself (Alighanbari and How (2005)). This approach is robust to malfunctions of the UAVs. The decentralized task assignment assumes exact situational awareness to make consensus among the UAVs. If the communication is not smooth, conflicts can occur among the mission plans of the UAVs. To overcome this weakness, Choi et al. (2009) suggested the consensus-based bundle algorithm (CBBA). In CBBA, the UAVs make consensus within certain predefined rules to avoid task conflicts among the UAVs, and it is advantageous in a situation involving various constraints. For example, Choi et al. (2010) extended CBBA for allocating heterogeneous tasks to UAVs that have different capabilities in a cooperative track and strike mission.

In this paper, an algorithm to improve the performance of existing CBBA in an engagement scenario is studied by considering the optimal scoring matrix, which reflects the heterogeneity between the UAVs and targets. Because both the UAVs and targets have different attack capabilities depending on the types, each of them has to choose the most suitable targets. Therefore, the scoring matrix linked to the performance of CBBA should be optimized. The algorithm to optimize the scoring matrix is the episodic parameter optimization (EPO) that utilizes reinforcement learning (RL) and particle swarm optimization (PSO) (Sutton and Barto (1998), Kennedy and Eberhart (1995)). The performance measure of CBBA considered here is the survivability of the UAVs. The EPO algorithm is performed while many episodes of the engagement are run repeatedly and the reward of each episode is updated using RL, and candidate scoring matrices are selected by using PSO. After several iterations of the EPO elapsed, the optimal scoring matrix is obtained for enhancing the performance of CBBA.

⋆ This research was financially supported by a grant to Unmanned Technology Research Center funded by Defense Acquisition Program Administration, and by Samsung Thales.

978-3-902823-37-3/2013 © IFAC

10.3182/20130703-3-FR-4038.00072

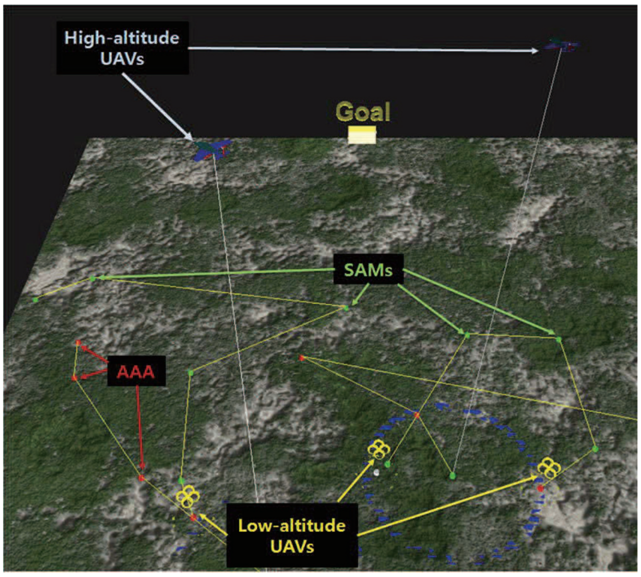

Fig. 1. The engagement scenario

The rest of this paper is organized as follows: Section 2 describes the basic problem setup about the scenario and models. The details of CBBA and the scoring matrix are introduced in section 3. Section 4 describes the parameter optimization algorithm to optimize the scoring matrix. The optimization results and discussions are given in section 5. The conclusions are presented thereafter.

2. PROBLEM FORMULATION

This section describes the basic settings for the engagement considered in this paper, and the control and guidance law of UAVs.

2.1 Scenario and Model Description

In the engagement scenario, a team of N_u UAVs are operated to overwhelm N_t stationary threats in a three-dimensional environment as shown in Fig. 1. The UAVs $i \in \{1, \dots, N_u\}$ are composed of two types of UAVs in which one is the low-altitude helicopter UAV ($k_i = 1$) and the other is the high-altitude fixed-wing UAV ($k_i = 2$). Here k_i denotes the type of the UAV i. The threats $j \in \{1, \dots, N_t\}$ also consist of the anti-aircraft artillery (AAA, $k_j = 1$) and the surface-to-air missiles (SAMs, $k_j = 2$). k_j denotes the type of the threat j. Assume that each UAV shares the information that contains the locations and the types of the threats with the other UAVs, and they are fully connected. Both the UAVs i and the threats j have different attack capabilities which are described by the attack probability, the attack range and the attack cycle depending on the type k_i and k_j. Therefore, the UAVs have to select the proper targets to maximize the team survivability while the UAVs pass over the target and overwhelm the threats.
The status Ξ_i of the UAV i consists of the set 3-tuples as follows:

$$\Xi_i = \langle \mathcal{X}_i, \mathcal{S}_i, \mathcal{A}_i \rangle \tag{1}$$

where $\mathcal{X}_i$ is the location of the UAV i, i.e., $(X_i, Y_i, h_i) \in \mathcal{X}_i$. $\mathcal{S}_i \in \{0, 1\}$ indicates the viability, and $\mathcal{A}_i \in \{0, 1\}$ denotes the possibility of the attack related to the attack cycle t_{ac}.

Table 1. Choices of A and B (UAV side)

	AAA ($k_j = 1$)		SAM ($k_j = 2$)	
	A	B	A	B
Low-altitude UAV ($k_i = 1$)	0.8	0.5	0	0.5
High-altitude UAV ($k_i = 2$)	0.4	0.2	0.8	0.2

Table 2. Choices of A and B (Threat side)

	AAA ($k_j = 1$)		SAM ($k_j = 2$)	
	A	B	A	B
Low-altitude UAV ($k_i = 1$)	0.8	0.1	0.4	0.5
High-altitude UAV ($k_i = 2$)	0	0.1	0.8	0.5

Similarly, the threat j also has the information of its own status $\Xi_j = \langle \mathcal{X}_j, \mathcal{S}_j, \mathcal{A}_j \rangle$.

2.2 Probabilistic Model

As mentioned in section 2.1, each UAV has different attack capabilities according to its type, so the probability for the UAV i to kill the threat j is defined by the following:

$$\begin{aligned} &P(\mathcal{S}_j = 0 | \mathcal{S}_j = 1, \mathcal{S}_i = 1, \mathcal{A}_i = 1) \\ &= \mathrm{Bern}(\mathcal{S}_j = 0 | \mu_i(X_j, Y_j, A, B)) \text{ for } k_i \text{ and } k_j \end{aligned} \tag{2}$$

where $\mathrm{Bern}(x|\mu) = (1-\mu)^x \mu^{(1-x)}$ is known as the *Bernoulli* distribution, and the gaussian function $\mu_i(\cdot,\cdot,\cdot,\cdot)$ is defined as follows.

$$\mu_i(x, y, a, b) = a \exp\left(-\left(\frac{(x - x_o)^2}{2b\sigma_x^2} + \frac{(y - y_o)^2}{2b\sigma_y^2}\right)\right) \tag{3}$$

Here $(x_o, y_o, \sigma_x, \sigma_y)$ are set as $(X_i, Y_i, \sqrt{h_i}, \sqrt{h_i})$. Both A and B are constants, and A is used to determine the striking power related to the interrelationships between UAVs and threats. B is for the attack range. Eq. (2) shows that when both the UAV i and the threat j are alive ($\mathcal{S}_i = 1, \mathcal{S}_j = 1$) and the UAV is ready to attack ($\mathcal{A}_i = 1$), the probability of kill is given by the Bernoulli distribution. As shown in Table 1 the low-altitude UAV ($k_i = 1$) can only contribute to eliminate the AAA ($k_j = 1$), so A is set to 0.8 for the AAA and 0 for the SAM. In the case of the high-altitude UAV ($k_i = 2$), they can exert influence on both the AAA ($k_j = 1$) and the SAM ($k_j = 2$) and have more contributions on the SAM than the AAA. Therefore, A is set to 0.4 for the AAA and 0.8 for the SAM. In addition, the possibility of attack $\mathcal{A}_i$ is determined by the following:

$$\mathcal{A}_i = \begin{cases} 1 & \text{if } t - t_a > t_{ac} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where t_a denotes a time to attack and is updated when the attack is begun at time t ($t_a = t$). The attack cycle t_{ac} is always positive.
Similar to (2)–(4), the threat j can attack the UAV i with the probability of kill which is defined by the following.

$$\begin{aligned} &P(\mathcal{S}_i = 0 | \mathcal{S}_i = 1, \mathcal{S}_j = 1, \mathcal{A}_j = 1) \\ &= \mathrm{Bern}(\mathcal{S}_i = 0 | \mu_j(X_i, Y_i, A, B)) \text{ for } k_i \text{ and } k_j \end{aligned} \tag{5}$$

For μ_j, $(x_o, y_o, \sigma_x, \sigma_y)$ are set as $(X_j, Y_j, \sqrt{r_j}, \sqrt{r_j})$ in (3). Here r_j is a constant for the attack range. As shown in Table 2, AAA ($k_j = 1$) can only attack the low-altitude UAV ($k_i = 1$) and the attack range is shorter than the SAM. The SAM can strike both the high- and low-altitude

UAVs and is more effective to counteract the high-altitude UAVs. If the UAV is destroyed by the attack of threat ($\mathcal{S}_i = 0$), the UAV cannot regenerate and move anywhere.

2.3 Control and Guidance Law of UAVs

The position controller of the low-altitude UAV is implemented by using an adaptive sliding mode controller (Lee et al. (2009)). The high-altitude UAV dynamics and control logic are applied by using the study of Kim and Kim (2009), and the trajectory generator is considered by the following:

$$\begin{aligned}\dot{X}_i^d &= V_i^d \cos\psi_i^d \\ \dot{Y}_i^d &= V_i^d \sin\psi_i^d \\ \dot{\psi}_i^d &= u_i \\ \dot{V}_i^d &= 0 \\ \dot{h}_i^d &= 0\end{aligned} \tag{6}$$

where (X_i^d, Y_i^d, h_i^d) is the desired inertial position of the i-th UAV, ψ_i^d is the desired heading, V_i^d is the desired velocity, h_i^d is the desired altitude. The desired altitude is always positive, and the desired heading rate input u_i is designed by using a limit cycle navigation (Kim and Kim (2003)). The details related to the low-level controls are omitted because this paper is more focused on the high-level control policy such as the mission assignment.

3. MISSION ASSIGNMENT

This section briefly describes the consensus-based bundle algorithm (CBBA) which is used for our mission (Choi et al. (2009)). Here, we name the UAV and the threat as an agent and a task, respectively.

CBBA is one of the decentralized task assignment method that is based on the market auction algorithm. Especially, CBBA considers agents' paths which are decided by their sequential task lists to solve the multi-assignment problem. The algorithm consists of two phases in each iteration. The first is bundle construction and the second is conflict resolution by local communication. Details of the algorithm are given in Algorithms 1 and 2.

In the first phase, each agent continuously adds tasks to its bundle in the order of decreasing reward of the task until it is incapable of adding any others. Each agent carries four vectors: a bundle $\boldsymbol{b}_i \in \mathbb{N}^{L_t}$, the corresponding path $\boldsymbol{p}_i \in \mathbb{N}^{L_t}$, a winning bid list $\boldsymbol{y}_i \in \mathbb{R}^{N_t}$, and a winning agent list $\boldsymbol{z}_i \in \mathbb{N}^{N_t}$. L_t and N_t denote the maximum number of tasks an agent can take and the total number of tasks, respectively. Tasks in the bundle $\boldsymbol{b}_i$ are ordered based on which ones were added first, while in the path $\boldsymbol{p}_i$ they are ordered based on their location in the assignment. $\boldsymbol{y}_i$ and $\boldsymbol{z}_i$ are the agent i's knowledge vectors which contain the knowledge about which agent takes each task (winning agent) and the successful bidding values (winning bid), respectively. In line 5 of Algorithm 1, $a \oplus_{end} b$ denotes the operation that inserts the list b after the last element of the list a. Every time the agent includes a new task to its bundle and path, it saves the knowledge about $\boldsymbol{y}_i$ and $\boldsymbol{z}_i$ (line 7 of Algorithm 1). In line 4 of Algorithm 1, when agent i takes the task j, the marginal score improvement $c_{i,j}$ according to the current path $\boldsymbol{p}_i$ is given as follows.

Algorithm 1 CBBA Phase 1: for agent i at iteration t

```
1: procedure BUILD BUNDLE ($\boldsymbol{z}_i(t-1), \boldsymbol{y}_i(t-1), \boldsymbol{b}_i(t-1)$)
2:     $\boldsymbol{z}_i(t) = \boldsymbol{z}_i(t-1)$; $\boldsymbol{y}_i(t) = \boldsymbol{y}_i(t-1)$; $\boldsymbol{b}_i(t) = \boldsymbol{b}_i(t-1)$;
3:     while $|\boldsymbol{b}_i(t)| < L_t$ do
4:         Find task $J_i$ which gives the most marginal score improvement, $c_{i,J_i}$ for given $\boldsymbol{b}_i(t)$, and $\boldsymbol{p}_i(t)$;
5:         $\boldsymbol{b}_i(t) = \boldsymbol{b}_i(t) \oplus_{end} \{J_i\}$;
6:         $\boldsymbol{p}_i(t) = \boldsymbol{p}_i(t) \oplus_{n_{i,J_i}} \{J_i\}$;
7:         $y_{i,J_i}(t) = c_{i,J_i}$; $z_{i,J_i}(t) = i$;
8:     end while
9: end procedure
```

Algorithm 2 CBBA Phase 2: agent i's action for task j

```
1: Local communication exchanging $(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{s})$ with agent $k$
2: Decide action by update rules shown in Choi et al. (2009).
3: 1) update : $y_{ij} = y_{kj}$, $z_{ij} = z_{kj}$;
4: 2) reset : $y_{ij} = 0$, $z_{ij} = 0$;
5: 3) leave : $y_{ij} = y_{ij}$, $z_{ij} = z_{ij}$;
```

$$c_{ij} = \begin{cases} \max\limits_{n \le |\boldsymbol{p}_i|+1} S_i^{\boldsymbol{p}_i \oplus_n \{J_i\}} - S_i^{\boldsymbol{p}_i} & \text{if } j \notin \boldsymbol{p}_i \\ 0 & \text{if } j \in \boldsymbol{p}_i \end{cases} \tag{7}$$

where $|\cdot|$ denotes the cardinality of the list, and $a \oplus_n b$ denotes the operation that inserts the list b right after the n-th element of the list a. Let $S_i^{\boldsymbol{p}_i}$ be defined as the total reward value for agent i performing the task along the path $\boldsymbol{p}_i$. In this paper the value of $S_i^{\boldsymbol{p}_i}$ is inversely proportional to the distance of the path as in example below.

$$S_i^{[1,2,3]} = \frac{m_{k_i,k_1}}{d_{i,1}} + \frac{m_{k_i,k_2}}{d_{i,1} + d_{1,2}} + \frac{m_{k_i,k_3}}{d_{i,1} + d_{1,2} + d_{2,3}} \tag{8}$$

where m_{k_i,k_j} denotes a element of the scoring matrix $M \in \mathbb{R}^{2\times 2}$ varying with the type k_i and k_j of the UAV i and the threat j. The scoring matrix M is used to reflect the hostile relationship between the UAVs and the targets, with m_{k_i,k_j} representing the relative strength of the agent i when confronting target j. $d_{a,b}$ denotes the distance from UAV a to threat b or threat a to threat b. For better performance related to the increase in team survivability, the scoring matrix M has to be selected optimally.

In the conflict resolution phase, three vectors are communicated for consensus. Two are the winning bid list $\boldsymbol{y}_i$ and the winning agent list $\boldsymbol{z}_i$, and the third vector $\boldsymbol{s}_i \in \mathbb{R}^{N_u}$ represents the time stamp of the last information update from each of the other agents, where N_u denotes the total number of agents. When agent i receives a message from agent k, $\boldsymbol{z}_i$ and $\boldsymbol{s}_i$ are used to determine which agent's information contains the most recent data for each task. There are three possible actions agent i can do on task j as shown in lines 3–5 of Algorithm 2. Detailed action rules during communication are given in Choi et al. (2009).

The CBBA process with a synchronized conflict resolution phase over a static communication network with diameter D guarantees at least 50% optimality in performance with in $N_{\min}D$ convergence time, where $N_{\min} = \min\{L_t N_u, N_t\}$. As mentioned in section 2.1, we assume that all UAVs are fully connected in communication. However, if some UAVs are destroyed, then they lose connections and the remaining UAVs acquire new mission points after performing CBBA except the UAVs that lost mission capability.

4. OPTIMIZATION FOR MISSION ASSIGNMENT

The scoring matrix presented in section 3 requires the proper selection of each element. This section describes the reinforcement learning and the particle swarm optimization to learn the best values for the elements of the scoring matrix.

4.1 Reinforcement Learning

Reinforcement learning can provide solutions in the situation involing the probabilistic model for the attack (Sutton and Barto (1998)). For a specific set of the scoring matrices, the performance of each scoring matrix is evaluated by the expected return. Then their suboptimal values of elements of the scoring matrix will be determined using episodic optimization.

The target assignment depends on the scoring matrix M and can be considered a function of M, so the assignment policy is denoted by $\pi(M)$. In the considering scenario, the main purpose of the UAVs is to maximize their survivability during the engagement by using the policy $\pi(M)$. Given initial conditions of an engagement, the expected return $V^{\pi(M)}$ is assigned to $\pi(M)$:

$$V^{\pi(M)} = E[R|\pi(M)] \tag{9}$$

where the random variable R denotes the return and is defined by the ratio of the survived UAVs:

$$R = \frac{N_s}{N_u} \tag{10}$$

where N_s is the number of the survived UAVs.

Because the exact evaluation of (9) is intractable (Sutton and Barto (1998)), instead the averaged total reward over the N_{ep} repeated runs is used by the following:

$$\mathcal{R} = \frac{1}{N_{ep}} \sum_{n_{ep}=1}^{N_{ep}} R^{(n_{ep})} \tag{11}$$

Here, N_{ep} is the total number of monte-carlo type simulations called episodes. At the end of each n_{ep}-th episode, the return $R^{(n_{ep})}$ is obtained. This process is repeated $n_{ep} = 1, \ldots, N_{ep}$ times, then their average $\mathcal{R}$ is computed according to (11) as shown in the lines 7–13 of Algorithm 3. This $\mathcal{R}$ will give an estimated performance for the particular values of elements of scoring matrix M. The performance estimates $\mathcal{R}'$ for different parameter values M' can be computed using the same process, and this process is repeated until convergence to the best values of M. To guarantee the sub-optimal property of the resulting parameters, the total number of episodes N_{ep} should be large enough (roughly speaking, some polynomial of the *complexity* of the problem described in Ng and Jordan (2000)), and the *same* set of random elements needs to be used to evaluate different values of parameters. In the considered problem, the averaged total reward (11), i.e., survivability, should be maximized for a candidate scoring matrix.

Table 3. Inputs of the EPO algorithm

Parameter	Explanation	Value
S	total number of particles $(1 < s < S)$	10
lb	lower bound of m_{k_i,k_j}	0
ub	upper bound of m_{k_i,k_j}	10
N_{ep}	total number of episodes $(1 < n_{ep} < N_{ep})$	100
$It_{\max}$	maximum iteration $(1 \leq it \leq It_{\max})$	50

Algorithm 3 Episodic Parameter Optimization (EPO)

```
 1: procedure RETURN OPTIMAL SCORING MATRIX ($M^*$)
 2:     Setup $N_{ep}$ initial conditions of the engagement
 3:     Initialize particles ($M_s, s = 1, \ldots, S$) with random position and velocity matrices
 4:     Initialize the $pBest_s$ and the value function for each particle $s$, i.e., $pBest_s \Leftarrow M_s$ and $\mathcal{R}_s^{(0)} \Leftarrow -\infty$
 5:     for $it \leq It_{\max}$ do
 6:         for each $s = 1, \ldots, S$ do
 7:             for $n_{ep} \leq N_{ep}$ do
 8:                 Start from the $n_{ep}$-th initial conditions
 9:                 Run episode $n_{ep}$ of the engagement
10:                 Update the rewards $R_s^{(n_{ep})}$
11:                 $n_{ep} \Leftarrow n_{ep} + 1$
12:             end for
13:             Compute $\mathcal{R}_s^{(it)}$ using (11)
14:             if $\mathcal{R}_s^{(it)} > \mathcal{R}_s^{(it-1)}$ then
15:                 $pBest_s \Leftarrow M_s$
16:             else
17:                 $\mathcal{R}_s^{(it)} \Leftarrow \mathcal{R}_s^{(it-1)}$
18:             end if
19:             $V^{\pi(M_s)} \Leftarrow \mathcal{R}_s^{(it)}$
20:         end for
21:         Set the best of $pBests$ as $gBest$,
22:         i.e., $gBest \Leftarrow \arg\max_{M_s} V^{\pi(M_s)}, s = 1, \ldots, S$
23:         Update each particle's velocity and position by (12)
24:         $it \Leftarrow it + 1$
25:     end for
26:     return $M^* \Leftarrow gBest$
27: end procedure
```

4.2 Particle Swarm Optimization (PSO)

To solve the parameter optimization problem, particle swarm optimization technique is employed. PSO is a population-based stochastic optimization technique (Kennedy and Eberhart (1995)). Every particle in the population travels in the search space looking for the global minimum (or maximum) similar to the behavior of bird flocking. Each particle adjusts its velocity according to its own experience and its swarm's experience while particles search for the global solution.

To determine the optimal scoring matrix M^*, the particles are regarded as the candidate scoring matrices. All the particles in the population which begin with a random position M_s and random velocity $v_s \in \mathbb{R}^{2\times2}$, $s = 1, \ldots, S$ where S is the swarm size, are candidate solutions and iteratively move in the problem space. The best previous position of the particle s is remembered and represented as $pBest_s \in \mathbb{R}^{2\times2}$. The position of the best particle among all the particles is represented as $gBest \in \mathbb{R}^{2\times2}$. At each iteration, the velocity v_s and position M_s of each particle s can be updated by the following.

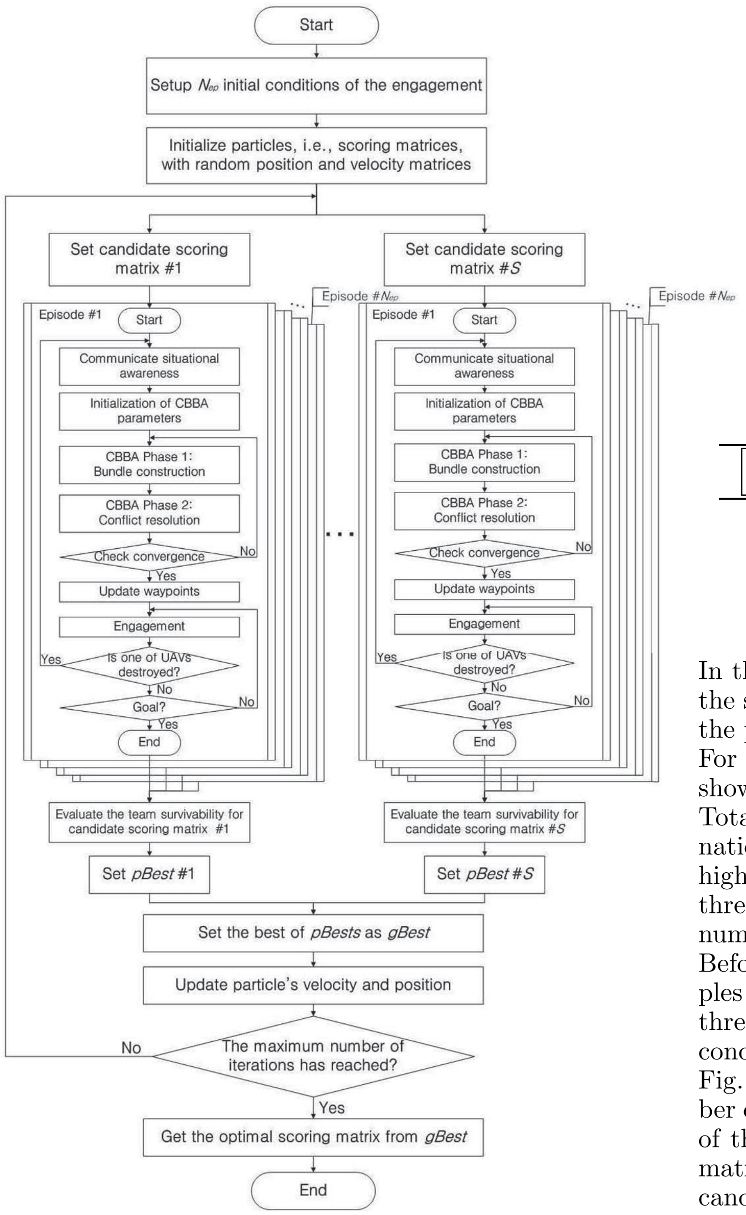
Start
Setup N_{ep} initial conditions of the engagement
Initialize particles, i.e., scoring matrices, with random position and velocity matrices
Set candidate scoring matrix #1
Set candidate scoring matrix #S
Episode #1
Episode #N_{ep}
Start
Communicate situational awareness
Initialization of CBBA parameters
CBBA Phase 1: Bundle construction
CBBA Phase 2: Conflict resolution
Check convergence
No
Yes
Update waypoints
Engagement
Is one of UAVs destroyed?
Yes
No
Goal?
No
Yes
End
Evaluate the team survivability for candidate scoring matrix #1
Evaluate the team survivability for candidate scoring matrix #S
Set pBest #1
Set pBest #S
Set the best of pBests as gBest
Update particle's velocity and position
The maximum number of iterations has reached?
No
Yes
Get the optimal scoring matrix from gBest
End

Fig. 2. Flow chart of the EPO algorithm

$$\begin{aligned} v_s &= K[v_s + c_1 r_1(pBest_s - M_s) + c_2 r_2(gBest - M_s)] \\ M_s &= M_s + v_s \end{aligned} \tag{12}$$

where c_1 and c_2 are the acceleration constants, and r_1 and r_2 are chosen as uniform random values in the range $[0,1]$, K is the constriction factor to ensure the convergence of PSO (Clerc and Kennedy (2002)), and it is determined by

$$K = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \tag{13}$$

where $\varphi = c_1 + c_2, \varphi > 4$. Typically, φ is set to 4.1. Then, the value function $V^{\pi(M_s)}$ determined by the scoring matrix M_s for each particle s can be evaluated in each PSO iteration it. The details of this optimization process are shown in Algorithm 3 and Fig. 2, and the inputs of the parameter optimization algorithm are set as shown in Table 3.

Table 4. Simulation Parameters

Parameter	Value
V_i^d	140 km/h
h_i^d $(k_i = 1)$	500 m
h_i^d $(k_i = 2)$	5 km
r_j $(k_j = 1)$	100 m
r_j $(k_j = 2)$	500 m
t_{ac} (UAV)	5 sec
t_{ac} (threat)	10 sec

Table 5. Optimized Scoring Matrices for different number of targets

$N_t = 15$	$N_t = 20$	$N_t = 25$
$\begin{bmatrix} 3.40575 & 0.128258 \\ 2.22477 & 5.65222 \end{bmatrix}$	$\begin{bmatrix} 9.55819 & 0 \\ 4.05476 & 9.7347 \end{bmatrix}$	$\begin{bmatrix} 2.95289 & 0 \\ 5.94118 & 10 \end{bmatrix}$

5. OPTIMIZATION RESULTS

In this section, the EPO algorithm is applied to optimize the scoring matrix M which is involved in the CBBA, and the performance of the proposed algorithms are analyzed. For a 10 km × 10 km × 10 km environment, the parameters shown in Table 4 remain fixed throughout all simulations. Total number of UAVs N_u is fixed at 5 and the combination of the UAVs is set as three low-altitude and two high-altitude UAVs. To observe the effect of the number of threats, three different scenarios are considered with the number of threats N_t set to 15, 20 and 25, respectively. Before the start of the EPO algorithm, $N_{ep} = 100$ samples of initial conditions about the locations and types of threats are uniformly randomly selected, and these initial conditions are applied for each episode.

Fig. 3 shows change in the team survivability as the number of iterations increases with the three different numbers of threats. At each EPO iteration, ten candidate scoring matrices are selected and the team survivability for each candidate is evaluated as shown in Fig. 3. All the cases show similar tendency in that the survivability improves after a few iterations. In each case, the survivability of end-stage is approximately 35% up from the initial value for $N_t = 15$. For $N_t = 20$ and $N_t = 25$ the chances of survival are enhanced by about 25% and 15%, respectively.

As a result, the EPO algorithm presents the positive effect to enhance the team survivability by optimizing the scoring matrix. Table 5 tells the validity of this conclusion. As mentioned in section 3, the rows of the scoring matrix M are related to the type of UAVs and the columns are for the type of threats. All the cases of the optimized scoring matrix have similar results in which $m_{1,2}$ is close to 0. This result means that the low-altitude UAV $(k_i = 1)$ will not choose SAMs $(k_j = 2)$ and satisfies our assumption in which the low-altitude UAV can only handle the AAA $(k_j = 1)$. The other example is that the value of $m_{2,2}$ is greater than the other elements because the high-altitude UAV $(k_i = 2)$ can only deal with the SAMs. Therefore, we can conclude that the scoring matrix is properly optimized by the EPO algorithm. CBBA with the optimized scoring matrix can allocate the heterogeneous targets to the UAVs by considering the UAVs' attack capabilities.

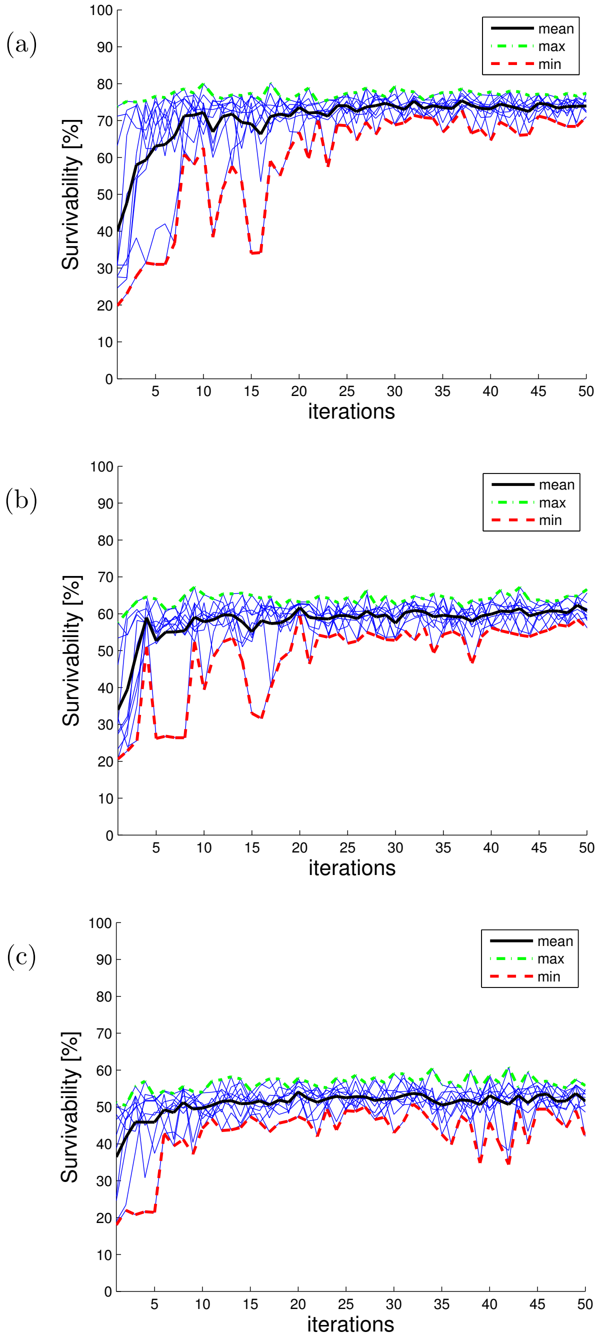

Fig. 3. Change in the survivability as the number of iterations increases (a) $N_t = 15$, (b) $N_t = 20$, (c) $N_t = 25$

6. CONCLUSION

This paper investigated the optimization of the decentralized task assignment for multiple heterogeneous UAVs in the probabilistic engagement scenario. Each UAV with the different attack capability can appropriately choose targets by employing a consensus-based bundle algorithm (CBBA) with the optimized scoring matrix that reflects heterogeneity among the UAVs and targets. The optimization of the scoring matrix related to the performance of CBBA was performed by employing an episodic parameter optimization (EPO) which uses reinforcement learning and particle swarm optimization. The optimization results showed that the team survivability of the UAVs is increased after performing the EPO algorithm and the values of the optimized score matrix are also optimally selected.

REFERENCES

Alighanbari, M. and How, J. (2005). Decentralized task assignment for unmanned aerial vehicles. In *Proc. 44th IEEE Conference on Decision and Control-European Control Conference (CDC-ECC '05)*.

Beard, R., McLain, T., Goodrich, M., and Anderson, E. (2002). Coordinated target assignment and intercept for unmanned air vehicles. *IEEE Trans on Robotics and Automation*.

Bellingham, J., Tillerson, M., Richards, A., and How, J.P. (2001). Multi-task assignment and path planning for cooperating uavs. In *Proc. Conference on Coordination, Control and Optimization*.

Choi, H., Brunet, L., and How, J. (2009). Consensus-based decentralized auctions for robust task allocation. *IEEE Trans on Robotics*.

Choi, H., Whitten, A., and How, J. (2010). Decentralized task allocation for heterogeneous teams with cooperation constraints. In *American Control Conference (ACC)*.

Chung, T. and Burdick, J. (2012). Analysis of search decision making using probabilistic search strategies. *IEEE Transactions on Robotics*.

Clerc, M. and Kennedy, J. (2002). The particle swarm explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans on Evolutionary Computation*, 6(1), 58–73.

Jin, Y., Minai, A., and Polycarpou, M. (2003). Cooperative real-time search and task allocation in uav teams. In *Proc. 42nd IEEE Conference on Decision and Control.*

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Proc. the 1995 IEEE International Conference on Neural Networks*, 1942–1948.

Kim, D. and Kim, J. (2003). A real-time limit-cycle navigation method for fast mobile robots and its application to robot soccer. *Robotics and Autonomous Systems*, 42(1), 17–30.

Kim, S. and Kim, Y. (2009). Optimum design of three-dimensional behavioural decentralized controller for uav formation flight. *Engineering Optimization*, 41(3), 199–224.

Lee, D., Kim, H., and Sastry, S. (2009). Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter. *International Journal of Control, Automation, and Systems*, 7(3), 419–428.

Ng, A. and Jordan, M. (2000). Pegasus: a policy search method for large mdps and pomdps. In *Proc. the 16th Conference on Uncertainty in Artificial Intelligence (UAI'00)*, 406–415.

Sutton, R. and Barto, A. (1998). *Reinforcement learning: an introduction.* MIT Press, Cambridge, Mass.

Yao, C., Ding, X.C., and Cassandras, C. (2010). Cooperative receding horizon control for multi-agent rendezvous problems in uncertain environments. In *Proc. 49th IEEE Conference on Decision and Control (CDC)*.