

Research Article

Modeling Multioperator Multi-UAV Operator Attention Allocation Problem Based on Maximizing the Global Reward

Yuhang Wu, Zhonghua Huang, Yinlin Li, and Zhiqi Wang

School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China

Correspondence should be addressed to Yinlin Li; liylinlin@bit.edu.cn

Received 15 June 2016; Revised 22 October 2016; Accepted 13 November 2016

Academic Editor: Asier Ibeas

Copyright © 2016 Yuhang Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper focuses on the attention allocation problem (AAP) in modeling multioperator multi-UAV (MOMU), with the operator model and task properties taken into consideration. The model of MOMU operator AAP based on maximizing the global reward is established and used to allocate tasks to all operators as well as set work time and rest time to each task simultaneously for operators. The proposed model is validated in Matlab simulation environment, using the immune algorithm and dynamic programming algorithm to evaluate the performance of the model in terms of the reward value with regard to the work time, rest time, and task allocation. The result shows that the total reward of the proposed model is larger than the one obtained from previously published methods using local maximization and the total reward of our method has an exponent-like relation with the task arrival rate. The proposed model can improve the operators' task processing efficiency in the MOMU command and control scenarios.

1. Introduction

The mode of human in the loop is widely used in the command and control (C2) of unmanned aerial vehicles (UAVs) presently. With the development of autonomous control and artificial intelligence technology, the problems in the single operator controlling multi-UAV (SOMU) have been investigated [1–5]. Given the increasing complexity of the missions, paradigm is shifting from single operator managing single UAV (SOSU) or multiple UAVs (SOMU) towards the concept of multioperator managing multi-UAV (MOMU) [6]. By assigning multiple operators to multiple UAVs, the flexibility of human-system decision-making can be improved [6]. Therefore, there have been many researchers working on C2 of MOMU [7–12]. However, the increased size of information captured by UAVs will restrict the tasks that need to be accomplished in a timely manner [6]. In order to improve the task processing throughput in MOMU, it is of necessity to understand which tasks the operators should attend to and when they should, which is known as the attention allocation problem (AAP).

Srivastava et al. [13] pointed out that work time and rest time for each task should be allocated to the operator. Bertuccelli et al. [4] proposed a nonpreemptive scheduling

formulation for a single operator performing a search mission with multi-UAV in a time-constrained environment. Jian et al. [5] modeled a single operator AAP with one operator controlling multi-UAV. Srivastava et al. [13–16] proposed an optimization framework to solve the concerns about how much attention the operator should be allocated as well as where it should be allocated. Crandall et al. [17, 18] introduced an attention allocation method for one operator controlling multirobot, and they applied this method with different strategies. It was found that guiding operator attentional resources can more effectively exercise the operator's judgment and experience than dictating them. However, the methods for single operator AAP proposed above are not applicable for multioperator AAP. In multioperator AAP, not only the work time and rest time but also how the tasks would be allocated to the operators should be considered and dealt with.

A few studies were concerned about multioperator AAP. Verma and Rai [6] modeled MOMU operator AAP and balanced reward maximization and operators' workload minimization. But the work time of tasks of the model was fixed and a local optimization of one task had been achieved. Majji and Rai [19] further extended the research of Srivastava et al. [16] and developed an optimized solution to the

multioperator AAP, which obtained a local optimization of one operator. The aforementioned methods accomplished a local optimization in solving multioperator AAP, without considering the influence of wait time on the tasks. Our research further improved the study of some previous works [5, 13, 19] by taking the wait time into consideration for multioperator AAP. Targeting to maximize the global reward, a model of multioperator AAP based on maximizing global reward is developed.

The paper is organized as follows: Section 2 introduces the theory on attention allocation; Section 3 proposes the MOMU operator AAP model; the simulation and result are presented in Section 4; and finally Section 5 discusses the conclusion and further improvement.

2. Theory on Attention Allocation Problem

Multioperator AAP in MOMU is dominated by various factors such as operator state, task properties, and environment [19]. Environment is a complex factor due to its unpredictability; thus the two following main aspects are considered here: operator state and task properties. Operator state includes operator utilization ratio, operator skill level, and operator performance. Task properties include task starting time, latency penalty due to wait time, weight, complexity, and reward.

2.1. Operator State

2.1.1. Operator Utilization Ratio. Operator utilization ratio represents the busy level of one operator. The higher the utilization ratio is, the busier the operator will be.

The relation of operator utilization ratio and operator performance obeys the Yerkes-Dodson law [20], which is shown in Figure 1. According to the Yerkes-Dodson law, the performance of a human operator is unimodal, which can be expressed by an inverted-U function of the utilization ratio [13]. When an operator is within an appropriate range of utilization ratio, the performance of the operator will be relatively high. Otherwise, the performance will decrease. High utilization ratio will fatigue the operator while low utilization ratio will make the operator out of the control loop, which will lead to the decreasing situational awareness of the operator. As a result, both high utilization ratio and low utilization ratio will distract the operator's focus on the C2 of UAVs. In this way, the operator will have to take more time to deal with a task. Operator utilization ratio x can be defined by the following differential equation [13]:

$$\dot{x}(t) = \frac{b(t) - x(t)}{\delta}, \quad (1)$$

where

$$b(t) = \begin{cases} 1, & \text{busy at time } t \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

δ is a constant that depends on operator's sensitivity.

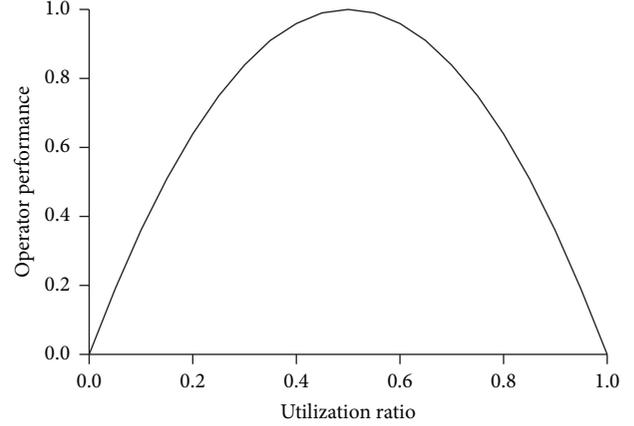


FIGURE 1: Yerkes-Dodson law.

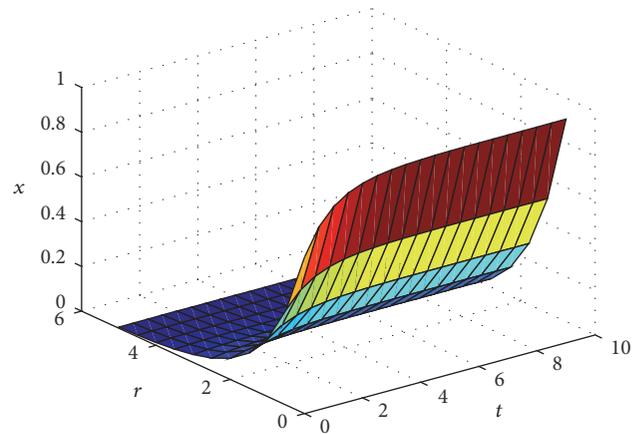


FIGURE 2: The evolution of operator utilization ratio with work time and rest time.

If the initial operator utilization ratio is $x_0 \in [x_{\min}, x_{\max}]$, the work time is t and the rest time after the task is r . Then, recursive expression for the operator utilization ratio is

$$x(t+r) = (1 - e^{-t/\delta} + x_0 e^{-t/\delta}) e^{-r/\delta}. \quad (3)$$

The evolution of operator utilization ratio with work time and rest time is shown in Figure 2. The utilization ratio will increase as the work time becomes longer. On the contrary, the ratio will decrease with longer rest time.

2.1.2. Operator Skill Level. Operator skill level represents the ability of the operator to process tasks. Operator skill level will affect the performance of operator. In this article, all operators are assumed to have the same skill level to avoid any difficulties in quantification.

2.1.3. Operator Performance. As UAVs are under supervisory control, a UAV operator can be treated as a two-alternative decision-making server. The decision made by an operator can be either correct or wrong. The probability of correct decision is defined to be the performance of the operator.

There are two models of two-alternative decision-making for operators.

Pew's Model [21]. The probability of correct decision at a given work time t , while $t \geq 0$, is expressed as follows:

$$P(H_1 | t) = \frac{p_0}{1 + e^{-(at-b)}}, \quad (4)$$

where H_1 is the correct decision, t is the work time, and p_0 , a , and b are the parameters related to operators and tasks. The evolution of the probability of the correct decision under Pew's model is shown in Figure 3(a).

Drift Diffusion Model [22]. The probability of correct decision at a given work time t , while $t \geq 0$, is expressed as follows:

$$P(H_1 | t) = \frac{1}{\sqrt{2\pi\sigma^2 t}} \int_{\eta}^{\infty} e^{-(\Lambda-\beta t)^2 / 2\sigma^2 t} d\Lambda, \quad (5)$$

where β is drift rate, σ is diffusion rate, η is decision threshold, and $\Lambda \equiv N(\beta t, \sigma^2 t)$ is the evidence at time t . The evolution of the probability of the correct decision under drift diffusion model is shown in Figure 3(b).

Pew's model has been widely used as operator performance models in recent researches on operator AAP [5, 16, 19]. In this paper Pew's model is also used as the operator performance model. The operator performance formula is

$$f(t) = \frac{p_0}{1 + e^{-(at-b)}}, \quad (6)$$

where a and b are parameters that are determined by operator skill level and task complexity. Assume that all the operators have the same skill level, so a and b are determined only by task complexity.

2.2. Task Properties

Task Weight. Tasks performed by UAV operators have different types. Even the tasks having the same type may have different degrees of importance. Thus, each task has a task weight denoted by w , which is used to represent the task importance [16].

Task Complexity. Task complexity is denoted by d , which can affect work time and operator performance.

Latency Penalty. When the task queue is not empty, operator cannot handle all the tasks on time. Tasks waiting in the queue will cause latency penalty. The latency penalty per unit time of one task is c [16] (each task has a different c); work time is t ; rest time after the task is r . All the tasks after the handling one requires wait time of $(t + r)$, which will cause latency penalty. Assume that the sum of latency penalty per unit time of all the tasks after the handling one is c' ; then the latency penalty caused by the handling task is $c' \cdot (t + r)$.

Task Reward. Task reward is the product of task weight and operator performance: $w \cdot f(t)$ [16].

Task Starting Time. As tasks arrive dynamically, their starting time is not the same, which causes different wait times in task queue. And different wait times will affect the latency penalty.

2.3. Evaluation Criterion of Attention Allocation

2.3.1. Integrated Reward. With dynamic arriving of tasks, the attention allocation is solved in multistage. In each stage, there is integrated reward of both a single task and a single operator.

Integrated reward of a single task is the task reward subtracting the latency penalty caused by this task. Integrated reward of a single operator is the sum of integrated rewards of all his current tasks in that stage.

2.3.2. Global Reward. Global reward is the evaluation criterion of attention allocation in one stage, which is the sum of integrated rewards of all the operators in one stage.

2.3.3. Total Reward. Total reward is the sum of global rewards in all stages, which is the evaluation criterion of attention allocation after all the tasks have been allocated.

3. MOMU Operator AAP Modeling

In Section 2, the concepts about attention allocation were introduced. According to the process of multi-UAV supervisory control, a multi-UAV multioperator attention allocation framework was established, which is shown in Figure 4.

UAV swarm generates and broadcasts the tasks, which need to be handled by the operators. And then, according to the task properties and operator state, the support system allocates all the tasks in the task queue to multiple operators using MOMU operator AAP model based on maximizing the global reward. At the same time, the support system allocates work time and rest time to each task. Operators deal with the tasks and send results to the support system to achieve the C2 of UAVs.

Assume that there are M operators, and they need to handle N tasks at some point; the operator performance for task $j \in \{1, 2, \dots, N\}$ is expressed by the operator performance formula (6): f_j . Task complexity is d_j , which is characterized by the pair (a_j, b_j) . According to the importance of the task, the support system will set a weight w_j to task j . Latency penalty per unit time of task j is c_j . Assume that the support system allocates work time t_j and rest time r_j to task j ; the reward of task j is $w_j f_j(t_j)$, and the latency penalty due to the wait time of task j is $(c_j + \dots + c_N)(t_j + r_j)$. The utilization ratio of operator i before processing task j is x_i^j . S_j is the function of x_i^j , which is used to determine the lower limit of expected work time of task j for operator i . The support system solves multioperator AAP in order to maximize the global reward at each stage.

At a stage, assume that operator i gets N' tasks. Work time $t_{j'}$ and rest time $r_{j'}$ are allocated to task j' , and task reward for processing task j' is $w_{j'} f_{j'}(t_{j'})$. The work time and rest time of task j' will not affect the tasks before it; the latency penalty caused by task j' is $(c_{j'} + \dots + c_{N'})(t_{j'} + r_{j'})$. Integrated reward of one task is the task reward subtracting the latency penalty caused by this task. Then, the integrated reward of task j' is

$$w_{j'} f_{j'}(t_{j'}) - (c_{j'} + \dots + c_{N'})(t_{j'} + r_{j'}). \quad (7)$$

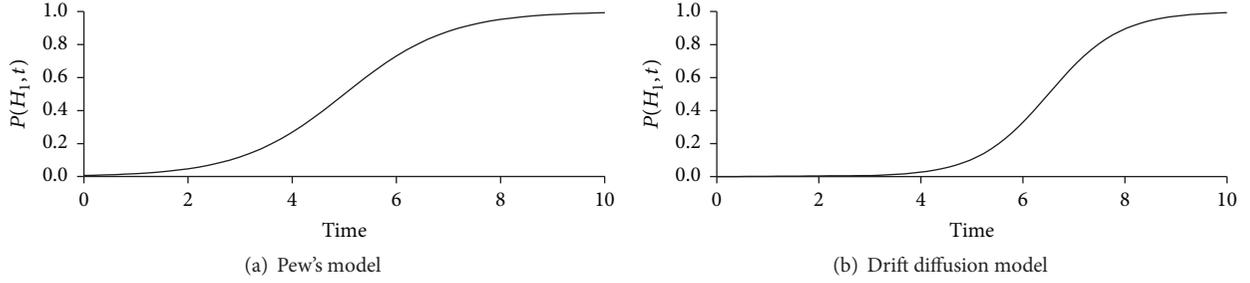


FIGURE 3: The evolution of the probability of the correct decision under Pew's and drift diffusion model.

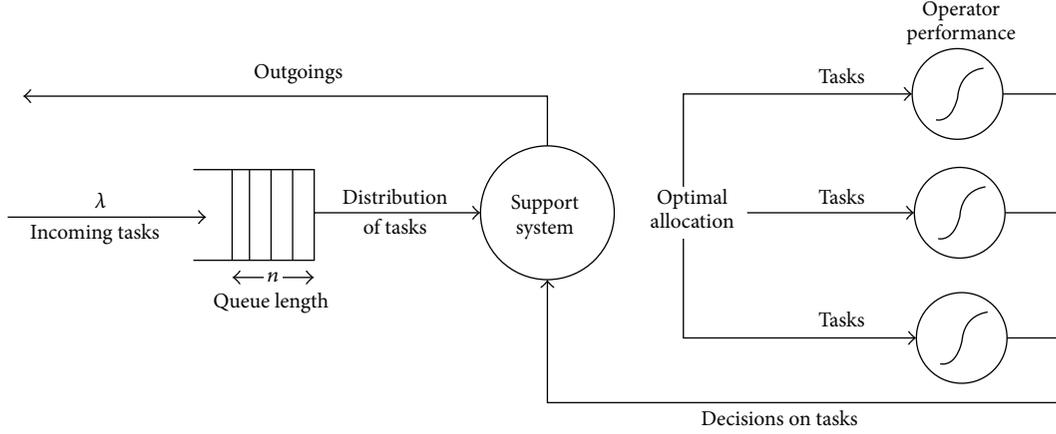


FIGURE 4: Multi-UAV multioperator attention allocation framework.

The integrated reward of operator i is the sum of integrated rewards of his N' tasks and it is written as

$$\sum_{j'=1}^{N'} [w_{j'} f_{j'}(t_{j'}) - (c_{j'} + \dots + c_{N'}) (t_{j'} + r_{j'})]. \quad (8)$$

The sum of integrated rewards of all the operators is the global reward. For all $j \in \{1, 2, \dots, N\}$, z_i^j indicates whether operator i is assigned to perform task j , where $z_i^j = 1$ means that operator i will process task j ; otherwise, operator i will not process task j . Task reward for processing task j is $z_i^j w_j f_j(t_j)$. The sum of latency penalty per unit time of all the tasks after task j is $(z_i^j c_j + \dots + z_i^N c_N)$; then latency penalty caused by task j is $z_i^j (z_i^j c_j + \dots + z_i^N c_N) (t_j + r_j)$. The integrated reward of task j of operator i is

$$z_i^j w_j f_j(t_j) - z_i^j (z_i^j c_j + \dots + z_i^N c_N) (t_j + r_j). \quad (9)$$

Then the integrated reward of operator i is

$$\sum_{j=1}^N [z_i^j w_j f_j(t_j) - z_i^j (z_i^j c_j + \dots + z_i^N c_N) (t_j + r_j)]. \quad (10)$$

Let x_i^j be the utilization ratio of operator i before starting task j and $S_j(x_i^j)$ be the function of x_i^j that captures expected service time of operator i on task j . The aim of the model is

to maximize the global reward in an attention allocation of a stage; the objective function of multioperator AAP, which is the sum of the integrated rewards of M operators, can be written as follows:

$$\max \quad J(\mathbf{t}, \mathbf{r}, \mathbf{z}) = \sum_{i=1}^M \sum_{j=1}^N [z_i^j w_j f_j(t_j) - z_i^j (z_i^j c_j + \dots + z_i^N c_N) (t_j + r_j)], \quad (11)$$

where $x_i^{j+1} = (1 - e^{-z_j t_j / \delta} + x_i^j e^{-z_j t_j / \delta}) e^{-z_j r_j / \delta}$, $z_i^j S_j(x_i^j) \leq z_i^j t_j$, $x_i^j \in [x_{\min}, x_{\max}]$, $z_i^j \in \{0, 1\}$, $t_j, r_j \in \mathbb{R} > 0$, $\forall j \in [1, N]$, $\forall i \in [1, M]$. $x_{\min}, x_{\max} \in [0, 1]$ are the bounds of operator utilization ratio, and \mathbf{t}, \mathbf{r} are N -vectors of t_j, r_j , which represent work time and rest time, respectively. \mathbf{z} is an $M \times N$ matrix of z_i^j , which represents task assignment indicator variable.

Suppose that tasks arrive at a certain rate λ ; the task queue will change dynamically. Attention allocation for dynamically arrived tasks will be solved at multistage. After an attention allocation which is referred to as a stage here, operators will work for a while to fulfill the tasks. And when one of the operators has no task to handle, the support system will start another stage. The interval between two adjacent stages is τ , which is the shortest total time of operators' remaining tasks in the previous stage. During time τ , the number of tasks in the task queue will be $N = \lambda \cdot \tau$. Assume that the number of the tasks will not change during a stage; at each stage the

number of tasks will be constant. Just before each stage, some of the operators' local task queues may not be empty, so two variables R_i and W_i are introduced. R_i is the integrated reward of remaining tasks in the local task queue of operator i , and W_i is the time required for the remaining tasks. The sum of latency penalty per unit time of all the new tasks is $\sum_{j=1}^N z_i^j c_j$, and the latency penalty cause by W_i is $W_i \cdot \sum_{j=1}^N z_i^j c_j$. Then the integrated reward of operator i is

$$R_i - W_i \cdot \sum_{j=1}^N z_i^j c_j + \sum_{j=1}^N [z_i^j w_j f_j(t_j) - z_i^j (t_j + r_j) (z_i^j c_j + \dots + z_i^N c_N)] \quad (12)$$

where $R_i = \sum_{j'=1}^{N_i} [w_{j'} f_j(t_{j'}) - (t_{j'} + r_{j'}) (c_{j'} + \dots + c_{N_i})]$, $W_i = \sum_{j'=1}^{N_i} (t_{j'} + r_{j'})$, and N_i is the number of remained tasks of operator i .

Two variables R_i and W_i are added to the integrated reward in (12). The objective function is the sum of the integrated rewards of M operators. Then the objective function is evolved to

$$\max J(\mathbf{t}, \mathbf{r}, \mathbf{z}) = \sum_{i=1}^M \left\{ R_i - W_i \cdot \sum_{j=1}^N z_i^j c_j \right.$$

$$\left. \sum_{l=1}^L \sum_{i=1}^M \left\{ R_i - W_i \cdot \sum_{j=1}^{N_i} z_i^j c_j + \sum_{j=1}^{N_i} [z_i^j w_j f_j(t_j) - z_i^j (t_j + r_j) (z_i^j c_j + \dots + z_i^{N_i} c_{N_i})] \right\} \right\} \quad (15)$$

4. Simulation

All the simulations are carried out in Matlab. The setup of each simulation of the method in this paper is shown in Simulation Setup.

Simulation Setup

- (1) Initialization: $M = 4$, $N = 10$, $Nt = 40$, $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{w}, \mathbf{x}, \mathbf{R} = 0$, $\mathbf{W} = 0$, $\lambda, l = 1$ (where Nt is the total number of the tasks);
- (2) Get first N properties of all the tasks remained. Delete the first N properties. $Nt = Nt - N$.
- (3) Generate n initial individuals: $\mathbf{z} \times n$;
- (4) Calculate fitness $J(\mathbf{t}, \mathbf{r}, \mathbf{z})$ using dynamic programming algorithm. Calculate the replication according to the Euclidean distance;
- (5) Termination condition is satisfied?

Yes: Output the optimal task assignment \mathbf{z}^* , \mathbf{t}^* , \mathbf{r}^* , go to step (6);

No: go to step (7);

- (6) if ($Nt > 0$)

$$+ \sum_{j=1}^N [z_i^j w_j f_j(t_j) - z_i^j (t_j + r_j) (z_i^j c_j + \dots + z_i^N c_N)] \left. \right\} \quad (13)$$

During an attention allocation, operator may be performing a task, and this task will not be a complete one. Assume that $\bar{t}_{j'}$ and $\bar{r}_{j'}$ are the remaining time to complete task j' ; then R_i and W_i are modified as

$$R_i = \sum_{j'=1}^{N_i} [\alpha w_{j'} f(t_{j'}) - (\bar{t}_{j'} + \bar{r}_{j'}) (c_{j'} + \dots + c_{N_i})], \quad (14)$$

$$W_i = \sum_{j'=1}^{N_i} (\bar{t}_{j'} + \bar{r}_{j'}),$$

where $\alpha = (\bar{t}_{j'} + \bar{r}_{j'}) / (t_{j'} + r_{j'})$, $\alpha \in [0, 1]$. Actually, only $\bar{t}_{j'}$ and $\bar{r}_{j'}$ of the first task in the local task queue of an operator are different from $t_{j'}$ and $r_{j'}$.

Assume that there are L stages; the total reward is the sum of the global rewards at all the stages. The global reward at each stage can be obtained by (13). At stage l there are N_l tasks. Then the total reward will be written as

Then $l = l + 1$, $N = \lambda \cdot \tau$; if $N > Nt$, then $N = Nt$. Calculate \mathbf{R} , \mathbf{W} using equation (14), go to step (2);

Else

Then end.

(7) Copy according to the fitness and concentration of replication;

(8) Cross;

(9) Variation, go to step (4).

4.1. The Result of MOMU Operator AAP Model Based on Maximizing Global Reward. Assume that per unit time one task comes into the task queue, as it means $\lambda = 1$. Task properties are randomly generated with bounds, where task complexity is a pair of random integers: $a \in [1, 6]$, $b \in [2, 36]$, latency penalty per unit time is $c \in [0, 0.3]$, and weight is a random integer of $w \in [1, 15]$. The lower limit of expected work time of task j for operator i is $S_j(x_i^j) = b_j(37x_i^{j^2} - 37x_i^j + 15)/6a_j$ [13]. The operator utilization ratio range is $x \in [0.5, 0.9]$. The number of operators is $M = 4$. The initial value of utilization of four operators is $[0.6, 0.8, 0.9, 0.7]$. The initial number of tasks in the task queue is 10, and there will be 30 tasks dynamically generated. The MOMU operator AAP model based on maximizing global reward is solved

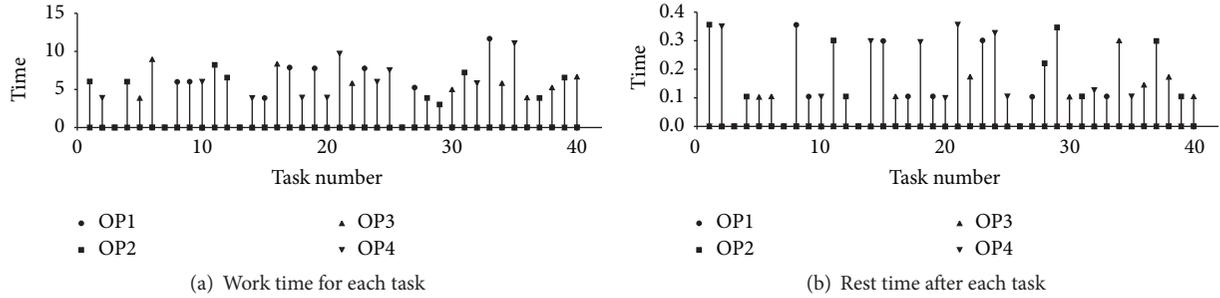


FIGURE 5: Result of attention allocation based on maximizing global reward.

by immune algorithm and dynamic programming algorithm. The result of this simulation is shown in Figure 5, where work time is shown in Figure 5(a) and rest time is shown in Figure 5(b).

The result of the simulation shows that the model can dynamically allocate the attention of operators. However, not all the tasks are handled during the simulation, while some tasks are dropped in order to maximize the global reward.

4.2. Comparison with Model Based on Maximizing Local Reward. This experiment is set to compare the model based on maximizing global reward (BOGR) with the model from the reference [19]. Since dynamic attention allocation was not introduced in [19], here is a brief description of dynamic attention allocation model based on maximizing local reward (BOLR).

Step 1. Sequence the operator in ascending order according to the utilization rate.

Step 2. Use the optimal solution from [19] to solve APP for all the operators in turn.

Step 3. Sequence the operator in ascending order according to the total time for all the tasks in their local task queues.

Step 4. Wait until one of the operators has no task. If the task queue is empty, wait a unit time for the tasks coming until no task comes; otherwise, return to Step 2.

With the initial number of tasks in the task queue being 10 and following 30 tasks dynamically generated, the model introduced in this paper which is based on maximizing global reward and the method described above which is based on maximizing local reward are used to solve multioperator AAP, respectively. In both methods, the total rewards are obtained. This experiment considers two conditions with the number of operators being $M = 2$ and $M = 4$, respectively. In these two conditions, ten times of simulations are carried out using the two methods. And the results are shown in Figures 6(a) and 6(b). It is shown that, compared to the total reward of the model based on maximizing local reward, the total reward obtained from the model in this paper is increased by 28% in the case of $M = 2$ on average while it is increased by 30% in the case of $M = 4$.

From the results of this experiment, it is shown that the total reward of the model based on maximizing global reward is larger than the reward obtained from the method based on maximizing local reward. But the model in this paper costs longer time than the compared model because of using intelligent optimization algorithm.

4.3. The Effect of Task Arrival Rate on Total Reward. The number of operators is 4 in this experiment, and the task arrival rate is from 1 to 10. For each arrival rate the simulation is carried out ten times, and the average values of the total reward for each arrival rate are shown in Figure 7.

The result shows that the total reward decreases with the increasing of the task arrival rate at the beginning ($\lambda < 5$). With the task arrival rate keeping increasing ($\lambda \geq 5$), the total reward fluctuates. The total reward decreases at the beginning ($\lambda < 5$), because tasks arrive earlier when the task arrival rate increases. Thus, they will have to wait longer in the task queue and task latency penalty will become larger. As the task arrival rate keeps increasing ($\lambda \geq 5$) and the total number of tasks remains constant, the support system will allocate all the tasks almost in two stages. The first stage will allocate 10 tasks and the second stage will allocate all 30 tasks dynamically generated. With the same number of stages, the same property of tasks will lead to nearly the same result, different properties of tasks will lead to different results without too many changes for the bounds of the properties. So the total reward fluctuates with no significant increase or decrease.

5. Conclusion

The MOMU operator AAP model based on maximizing the global reward is established in this paper. This model can dynamically allocate all the tasks in the task queue to the proper operators and set work time and rest time to each task at the same time. Validated by the simulation routine using Matlab environment, it is found that the total reward of our model is larger compared to the values obtained from the previous methods based on local reward maximization. Moreover, the result shows that the task arrival rate and total reward of MOMU operator AAP model have an exponent-like relation.

The proposed method improves the study of the MOMU operator attention allocation problem by evaluating the

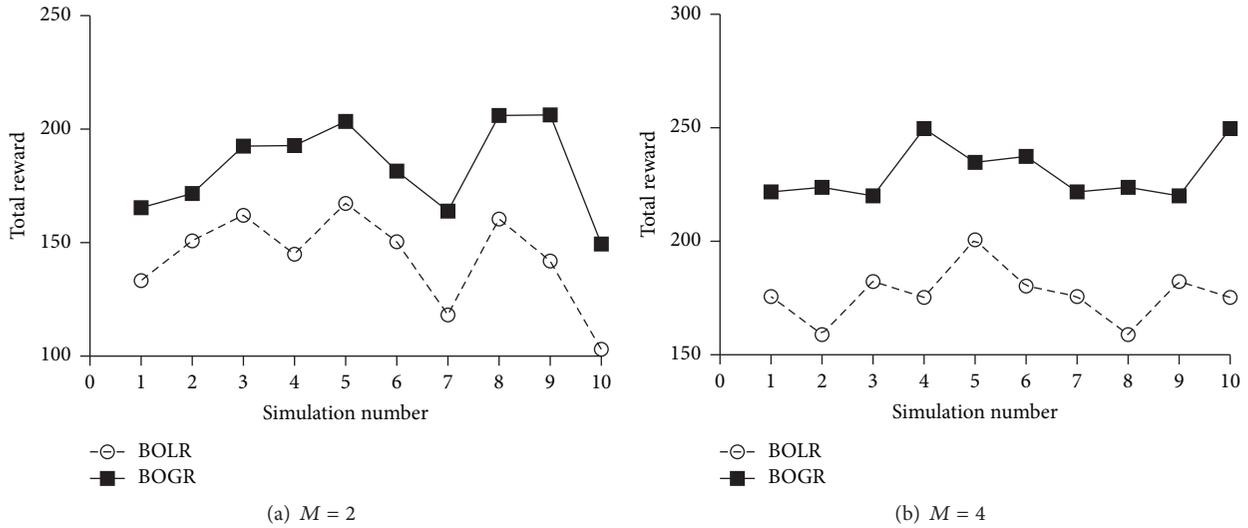


FIGURE 6: Results of comparison.

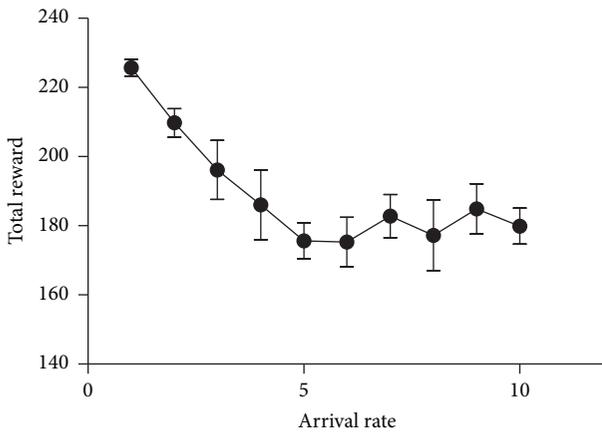


FIGURE 7: The relation of task arrival rate and total reward.

reward of task assignment planning of the command and control operator team, instead of only one single operator as concerned by the previous attempts. Accordingly, the team performance of command and control, or the global reward, has been enhanced. The method can be applied to the MOMU command and control scenarios and contribute to the task processing efficiency of the operator team.

In this paper, operators are treated as two-alternative decision-making models. However, in other cases, operators of UAVs may have more than two decisions for command and control. In the future, we will consider multialternative decision-making models for operators.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] D. Perez, I. Maza, F. Caballero, D. Scarlatti, E. Casado, and A. Ollero, "A ground control station for a Multi-UAV surveillance system: design and validation in field experiments," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 69, no. 1-4, pp. 119-130, 2013.
- [2] M. L. Cummings, S. Bruni, S. Mercier, and P. J. Mitchell, "Automation architecture for single operator, multiple UAV command and control," *International C2 Journal*, no. 2, 2007.
- [3] G. Zhang, X. Lei, Y. Niu, and D. Zhang, "Architecture design and performance analysis of supervisory control system of multiple UAVs," *Defence Science Journal*, vol. 65, no. 2, pp. 93-98, 2015.
- [4] L. F. Bertuccelli, N. W. M. Beckers, and M. L. Cummings, "Developing operator models for UAV search scheduling," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Toronto, Canada, August 2010.
- [5] L. Jian, D. Yin, L. Shen, and J. Yang, "Optimal attention allocation to visual search tasks of multi-UAVs based on operator model," in *Proceedings of the 12th IEEE International Conference on Mechatronics and Automation (ICMA '15)*, pp. 1767-1771, August 2015.
- [6] A. Verma and R. Rai, "Modeling multi operator-multi-UAV (MOMU) operator attention allocation problem," in *Proceedings of the ASME 2013 International Mechanical Engineering Congress and Exposition*, vol. 39, no. 1, San Diego, Calif, USA, November 2013.
- [7] B. Mekdeci and M. L. Cummings, "Modeling multiple human operators in the supervisory control of heterogeneous unmanned vehicles," in *Proceedings of the the 9th Workshop on Performance Metrics for Intelligent Systems*, pp. 1-8, Gaithersburg, Md, USA, September 2009.
- [8] F. Gao and M. Cummings, "Using discrete event simulation to model multi-robot multi-operator teamwork," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting (HFES '12)*, vol. 56, pp. 2093-2097, October 2012.
- [9] M. Lewis, H. Wang, S. Y. Chien, and P. Scerri, "Teams organization and performance in multi-human/multi-robot teams,"

- IEEE International Conference on Systems Man & Cybernetics*, vol. 12, no. 4, pp. 1617–1623, 2010.
- [10] P. J. Lee, H. Wang, S. Y. Chien, M. Lewis, and P. Scerri, “Teams for teams performance in multi-human/multi-robot teams,” *Human Factors & Ergonomics Society Annual Meeting Proceedings*, vol. 54, no. 4, pp. 438–442, 2010.
- [11] H. Wang, M. Lewis, and S.-Y. Chien, “Teams organization and performance analysis in autonomous human-robot teams,” in *Proceedings of the 10th Performance Metrics for Intelligent Systems Workshop (PerMIS '10)*, pp. 251–257, Baltimore, Md, USA, September 2010.
- [12] H. Wang, S. Y. Chien, M. Lewis, P. Velagapudi, P. Scerri, and K. Sycara, “Human teams for large scale multirobot control,” in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '09)*, pp. 1269–1274, October 2009.
- [13] V. Srivastava, A. Surana, and F. Bullo, “Adaptive attention allocation in human-robot systems,” in *Proceedings of the American Control Conference (ACC '12)*, pp. 2767–2774, Montreal, Canada, June 2012.
- [14] V. Srivastava, R. Carli, C. Langbort, and F. Bullo, “Task release control for decision making queues,” in *Proceedings of the American Control Conference (ACC '11)*, pp. 1855–1860, San Francisco, Calif, USA, June 2011.
- [15] V. Srivastava, A. Surana, M. P. Eckstein, and F. Bullo, “Mixed human-robot team surveillance,” *Mathematics*, In press.
- [16] V. Srivastava, R. Carli, C. Langbort, and F. Bullo, “Attention allocation for decision making queues,” *Automatica*, vol. 50, no. 2, pp. 378–388, 2014.
- [17] J. Crandall and M. Cummings, “attention allocation efficiency in human-UV teams,” in *Proceedings of the AIAA Infotech@Aerospace Conference*, vol. 17, pp. 1–9, Rohnert Park, Calif, USA, 2007.
- [18] J. W. Crandall, M. L. Cummings, M. D. Penna, and P. M. A. De Jong, “Computing the effects of operator attention allocation in human control of multiple robots,” *IEEE Transactions on Systems, Man, and Cybernetics C: Applications and Reviews*, vol. 41, no. 3, pp. 385–397, 2011.
- [19] M. Majji and R. Rai, “Autonomous task assignment of multiple operators for human robot interaction,” in *Proceedings of the American Control Conference*, pp. 6454–6459, 2013.
- [20] R. M. Yerkes and J. D. Dodson, “The relation of strength of stimulus to rapidity of habit-formation,” *Journal of Comparative Neurology and Psychology*, vol. 18, no. 5, pp. 459–482, 1908.
- [21] R. W. Pew, “The speed-accuracy operating characteristic,” *Acta Psychologica*, vol. 30, pp. 16–26, 1969.
- [22] R. Bogacz, E. Brown, J. Moehlis, P. Holmes, and J. D. Cohen, “The physics of optimal decision making: a formal analysis of models of performance in two-alternative forced-choice tasks,” *Psychological Review*, vol. 113, no. 4, pp. 700–765, 2006.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

