Research Article

Boolean Networks-Based Auction Algorithm for Task Assignment of Multiple UAVs

Xiaolei Sun, Naiming Qi, and Weiran Yao

School of Astronautics, Harbin Institute of Technology, Harbin 150001, China

Correspondence should be addressed to Xiaolei Sun; sun_xiaolei@msn.com

Received 5 February 2015; Revised 8 May 2015; Accepted 8 June 2015

Academic Editor: Fernando García

Copyright © 2015 Xiaolei Sun 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 presents an application of Boolean networks-based auction algorithm (BNAA) for task assignment in unmanned aerial vehicles (UAVs) systems. Under reasonable assumptions, the assignment framework consists of mission control system, communication network, and ground control station. As the improved algorithm of consensus-based bundle algorithm (CBBA), the BNAA utilizes a cluster-based combinatorial auction policy to handle multiple tasks. Instead of empirical method based on look-up table about conditional variables, Boolean network is introduced into consensus routine of BNAA for solving the conflict of assignment across the fleet of UAVs. As a new mathematic theory, semitensor product provides the implementation and theoretical proof of Boolean networks. Numerical results demonstrate the effectiveness and efficiency of proposed BNAA method.

1. Introduction

In the last decade, the unmanned aerial vehicles (UAVs) are developed into collaborative applications [1], such as search and destroy mission, persistent surveillance, border or perimeter patrol [2], and tracking and rescue mission [3]. These demanding applications call for an increasing level of autonomy for each one of the UAVs fleet, which must adapt to changes in situational awareness at real time [4]. As the number of UAVs and missions grows, the task assignment (TA) algorithms are considered to be NP-hard problem [5]. In recent works, this challenging problem is solved by many approaches, which include three main types of TA algorithms: centralized, decentralized, and auction algorithms [6].

Using central planner to communicate the central server that produces the plan for the fleet, centralized algorithm tends to yield a better solution than other approaches. However, in this architecture each UAV must consistently communicate with a fixed station, which greatly limits the scope of tasks that the fleet can execute. Moreover, in the mission, the centralized algorithm is subject to a single point failure [7]. Some decentralized approaches [8, 9] are proposed to eliminate above disadvantages. In order to achieve a conflict-free assignment, each UAV of the fleet must perform the same functionality as the central planner of centralized algorithm. For sharing the information of TA, it is generally assumed that the decentralized methods have the infinite bandwidth of communication. Thus, this approach may lead to an inefficient situation, where the member of the fleet will transfer a number of unnecessary data to each other. If that is not the case, although it can improve this inefficiency by introducing the consensus algorithm, decentralized approaches often require severe latency time to communicate [10]. More recently, auction algorithms based on market mechanism for TA are developed. It is effective for producing near-optimal solutions. After providing bids for match tasks, the UAV with highest value obtains corresponding assignment. To receive these bids and determine the highest value list, the traditional auction method utilizes an auctioneer as the central auction system [11]. Thus, these types of methods suffer the limitations of network topology since the UAV must communicate with the auctioneer for bid on the assignment. Even if the improved method removes the central system, it has been demonstrated that this method still has communication problems.

As an improved auction algorithm, the consensus-based bundle algorithm (CBBA) is developed by combining the
bundle method with the consensus approach. Many extended algorithms of CBBA demonstrate its advantage in TA field [6, 12–15]. According to the main components, there are two phases of CBBA. The bundle phase is utilized to the situation, where UAV will bid on multiple tasks for the improvement of the assignment efficiency. In addition, the consensus phase is addressed to solve the conflict of assignment between the UAV’s. The consensus algorithm of CBBA achieves by look-up table approach. It is essentially empirical method, although the original and extended CBBA methods can achieve the convergence to conflict-free assignment [14–16]. This manner cannot reveal the logical relation between the information vectors and decision actions.

Considering the efficiency of the computation and communication, this paper presents a Boolean networks-based auction algorithm (BNAA). As similar structure of CBBA, there are two phases for BNAA: combinatorial auction and Boolean networks-based consensus. In the first phase, the cluster method is utilized to assign the task combination adhering to a mission control systems integrated cooperating platform with the ability to perform the assignment efficiency. In addition, the consensus phase is introduced to model the consensus algorithm of look-up table. As an new way to realize the TA, Boolean networks can transfer conditional statements to mathematical theory based on new matrix product, namely, semitensor product [17]. Then, the Boolean network can be converted into an algebraic function, which is described via transition matrix [18]. Therefore, the consensus becomes the process of identification to the related matrices. In the framework of semitensor product approach, the realizability, controllability, observability, and disturbance decoupling of Boolean networks are also investigated in many papers [19–22].

The rest of this paper is organized as follows. Section 2 gives some preliminaries. Section 3 formulates the problem of TA for UAVs and provides the architecture of proposed method. In Section 4, the BNAA is discussed in detail. In Section 5, general scenarios present the performance of the BNAA through simulations results and Section 6 follows by some concluding remarks.

2. Preliminaries: Semitensor Product

As a new matrix product tool, semitensor product can convert a logical relation into an algebraic expression. By using this tool, the logical dynamics of a Boolean network can be transformed into discrete-time systems [22].

Definition 1. There are \( A_{\otimes_{\lor}} \) and \( B_{\otimes_{\land}} \), where \( z = \text{Mul}(n, w) \) denotes the least common multiple of \( n \) and \( w \). Then the semitensor product of \( A \) with \( B \) is defined as

\[
A \times B := \left( A \otimes I_{z/n} \right) \left( B \otimes I_{z/w} \right),
\]

where "\( \otimes \)" is the Kronecker product. If \( z = 1 \), then the semitensor product is the regular product of matrix. Thus, "\( \times \)" will be omitted in this paper.

First, we introduce some notations [19].

(i) \( \varphi : \Lambda^m \to \Lambda^k, \Lambda = \{0, 1\} \) is called logical function.

(ii) \( \delta^k \) denotes the \( k \)-th column of the identity matrix \( I_n \).

Particularly, \( \delta^1 = \{0, 1\}^T \).

(iii) \( \Delta_k = \{ \delta^k | k = 1, 2, \ldots, n \}, \Delta = \Delta_k \).

(iv) If \( M = [\delta^1_m; \delta^2_m; \ldots; \delta^p_m] \), then \( M \) is called a \( m \times p \)

logical matrix. It is briefly denoted as

\[
M = \delta [k_1 \ k_2 \ \cdots \ k_p].
\]

Particularly, if \( m = 2 \), then \( k_i \in \Lambda, i = 1, \ldots, p \). The binary number can be converted into a hexadecimal number as follows:

\[
M = \delta [k_1 \ k_2 \ \cdots \ k_p] = \delta [k_1 \ k_2 \ \cdots \ k_p]_{\text{hex}}.
\]

To utilize the matrix expression, we identify 0 as \( \delta^0 \) and 1 as \( \delta^1 \). Then the logical variable \( a_k = [a_k, 1-a_k]^T \in \Delta \) will be called vector-logical variable. The logical function \( \varphi : \Lambda^m \to \Lambda^k \) can be converted into a mapping \( \varphi : \Delta_m \to \Delta_k \). There is an important theorem, which is given by Theorem 11 in [20].

Theorem 2. Let \( f(a_1, \ldots, a_n) : \Lambda^m \to \Lambda^k \) be a logical function, which is converted into vector-logical form \( f : \Delta_m \to \Delta_k \).

Then there is a unique \( 2^m \times 2^n \) logical matrix \( M_f \), which is called the structure matrix of \( f(a_1, \ldots, a_n) \), such that

\[
f(a_1, \ldots, a_n) = M_f \otimes a = M_f \otimes M_{a_k}, \quad a_k \in \Delta.
\]

3. Task Assignment: Problem Statement

3.1. Task Assignment Architecture. In order to illustrate the function of TA, the integrated UAV’s system is shown in Figure 1. The assignment architecture of UAV addresses an integrated cooperating platform with the ability to perform the task assignment adhering to a mission control systems (MCS), for running collaborative and assignment algorithm. In this architecture, ground control station (GCS) is in charge of sending the instructions about taking-off and landing, loading the initial information about the mission and receiving the data from the sensors of UAVs. Each UAV’s MCS runs assignment algorithm and achieves the consensus of TA through the wireless communication network.

3.2. Task Assignment Scenario. Figure 2 illustrates the scenario of \( K \) tasks assignment for \( N \) UAVs. Let \( N = \{1, \ldots, N\} \) and \( K = \{1, \ldots, K\} \) be defined as the index sets for the UAVs and tasks, respectively. In this case, \( \text{UAV}_{n}, n \in \mathbb{N} \) represents the \( n \)-th UAV and \( \text{Task}_k, k \in K \) the \( k \)-th task. The general consideration is that one task is assigned to no more than one
UAV, that is, conflict-free assignment. The global objective function [6] of TA is given by

\begin{equation}
\max \sum_{n=1}^{N} \left( \sum_{k=1}^{K} R_{nk} \cdot h_{nk} \right),
\end{equation}

\begin{equation}
\text{s.t.} \quad \sum_{k=1}^{K} h_{nk} \leq N_n, \quad \forall n \in \mathbb{N},
\end{equation}

\begin{equation}
\sum_{n=1}^{N} h_{nk} \leq 1, \quad \forall k \in \mathbb{K},
\end{equation}

\begin{equation}
h_{nk} \in \{0, 1\}, \quad \forall (n, k) \in \mathbb{N} \times \mathbb{K},
\end{equation}

where score $R_{nk}$ is obtained by UAV$_n$ after receiving the assignment of Task$_k$, $h_{nk} \in \{0, 1\}$ indicates if UAV$_n$ is assigned to Task$_k$; $N_n$ is maximum number of tasks that UAV$_n$ can take; and $h_n = \{h_{n1}, \ldots, h_{nK}\}$ represents a vector of assignment for UAV$_n$. The solution of the TA depends heavily on the high accuracy model of score function. In the BNAA method, score function $R_{nk}$ consists of a task reward $J_k$ and executing cost $F_{nk}$ [1]. $R_{nk}$ is expressed as

\begin{equation}
R_{nk} = \Theta_{nk} (J_k - F_{nk}),
\end{equation}

\begin{equation}
J_k = g_k \cdot e^{-\varepsilon_k t},
\end{equation}

\begin{equation}
F_{nk} = \omega_n \cdot t,
\end{equation}
where $t$ is execution time; $g_k$ denotes a static score of Task $k$; $\xi_k$ and $\omega_n$ are weighting factors. $\Theta_{nk} = \beta_n \cdot \eta_{nk} \cdot \xi_k$, $\xi_k$ is constraint factor, which is defined as

$$\beta_n = \begin{cases} 1, & D_r(n) \geq D_n, \\ 0, & \text{otherwise}, \end{cases}$$

$$\eta_{nk} = \begin{cases} 1, & \text{UAV}_n \text{ can carry out Task}_k, \\ 0, & \text{otherwise}, \end{cases}$$

$$\xi_k = \begin{cases} 1, & c_k \leq t \leq d_k, \\ 0, & \text{otherwise}, \end{cases}$$

$$\xi_k = \begin{cases} 1, & t + \tau_k \leq d_k, \\ 0, & \text{otherwise}, \end{cases}$$

where $\beta_n$ denotes constraint of fuel level that can be provided to perform the rest distance $D_r(n)$, which must be greater than remaining distance of flight path $D_n$; $\eta_{nk}$ is the constraint about suitability of UAV$_n$ to Task$_k$; $\xi_k$ denotes the constraint of the execution time, where $c_k$ and $d_k$ describe the start time and end time of Task$_k$, respectively; $\xi_k$ represents the validity of Task$_k$, where $\tau_k$ is the duration in which the task must be performed.

4. BNAA for Task Assignment

The BNAA method is developed to assignment, in which each UAV will bid on a sequence of multiple tasks by introducing the combinatorial auction algorithm. The Boolean networks-based consensus algorithm is applied for dealing with the conflict of TA in the fleet. Each UAV stores its own information and updates it in continuous loops with these two phases. There are some vectors of information that UAV$_n$ must carry in the BNAA method [13]:

1. A matrix of tasks information $T = \{T_1, \ldots, T_K\}$ with vector $T_k = \{g_k, \epsilon_k, \omega_{nk}\}$, $k \in K$, in which all elements belong to $[0, \infty)$.

2. A highest bid list $H_n = \{H_{n1}, \ldots, H_{nK}\}$, where each element $H_{nk}$ represents the highest bid on Task$_k$ according to the information of UAV$_n$.

3. A creation time list of highest bid $b_k = \{b_{n1}, \ldots, b_{nk}\}$, where $b_{nk} \in [0, \infty)$ is the corresponding creation time of the element in $H_n$.

4. A winner list $W_n = \{W_{n1}, \ldots, W_{nk}\}$, in which element $W_{nk} \in \{N \cup \emptyset\}$ is the index of the UAV that wins Task$_k$ according to the information of UAV$_n$.

5. A combinatorial tasks list $Z_n = \{Z_{n1}, \ldots, Z_{n, N(n)}\}$, where $Z_{ni} \in K$ for $i \in \{1, \ldots, N(n)\}$, $N(n) \leq N_n$ represents the number of multiple tasks that UAV$_n$ has won. The elements of $Z_n$ are ordered according to the adding time of the tasks.

6. An execution sequence list $L_n = \{L_{n1}, \ldots, L_{n, N(n)}\}$, where $L_{ni} \in K$ indicates the order UAV$_n$ will perform the task in its combinatorial tasks list, which has the same tasks as $Z_n$.

(7) A time list of execution $s_n = \{s_{n1}, \ldots, s_{n, N(n)}\}$, in which $s_{ni} \in K$ is the corresponding time when UAV$_n$ will perform the task in $L_n$.

4.1. Combinatorial Auction. As shown in Figure 3, combinatorial auction process of BNAA is performed by using cluster algorithm [23], where all tasks are taken into account at each time step. Thus, the task combination with higher score has the priority to be assigned. The cluster algorithm has the following steps.

1. Computation of Distance between Classes. As each task is a class and assuming $D_k$, $k \in K$ denotes the $k$th class, and the distance matrix between classes is given by

   $$p^{(0)} = \begin{bmatrix} 0 & p_{21} & \cdots \\ p_{12} & 0 & \cdots \\ \vdots & \vdots & \ddots \\ p_{K1} & p_{K2} & \cdots & 0 \end{bmatrix}_{K \times K},$$

   where $p_{ij}$ represents the distance between Task$_i$ and Task$_j$.

2. Modification of Task Score Function. Given upper limit $p_{\text{max}}$ of the distance, if the minimum element in $p^{(0)}$ besides
0 is \( p_{ij} \leq p_{\text{max}} \), \( D_i \) and \( D_j \) are merged into a single class. For \( \text{UAV}_n \), the score of Task \( k \) is thereby modified as
\[
R'_n(t) = R_n(t) + y R_{nj} \left( t + \frac{p_{ij}}{V'} \right),
\]
where \( y \) is weighting factor.

(3) Update of Distance Matrix. According to modified information, the matrix is updated to \( p^{(i)} \). Steps (1) to (3) are repeated until \( p_{ij} > p_{\text{max}} \).

The local objective function \([6]\) of \( \text{UAV}_n \) is defined as
\[
V_n(L_n) = \sum_{k=1}^{K} R'_{nk} \cdot h_{nk}, \quad (10)
\]
which represents the sum of the rewards that \( \text{UAV}_n \) received by performing the tasks along \( L_n \). The marginal score function about combinatorial tasks \( Z_n \) \([13]\), that is, the improvement of the objective function when Task \( k \) is added, is given by
\[
r_{nk}(Z_n) = \begin{cases} 
\max_{i \in N(n)} V_n(L'_n) - V_n(L_n), & k \notin Z_n, \\
0, & k \in Z_n, 
\end{cases}
\]
\[
r_{nk}(Z_n) = \begin{cases} 
\max_{i \in N(n)} V_n(L'_n) - V_n(L_n), & k \notin Z_n, \\
0, & k \in Z_n, 
\end{cases}
\]
where \( L'_n = L_n \cup \{k\} \), the symbol \( \cup j \) is the operator that provides new set \( \{k\} \) after the \( i \)-th element of the \( L_n \), and \( \cup \text{end} \) indicates to add the elements at the end of first list. \( Z_n \) and \( L_n \) update as the following equation:
\[
Z_n = Z_n \cup \{k^*\},
\]
\[
L_n = L_n \cup \{k^*\},
\]
where
\[
k^* = \arg \max_{k} r_{nk}(Z_n) \cdot l_{nk},
\]
\[
i^* = \arg \max_{i} V_n(L'_n),
\]
where \( l_{nk} = 1 \) if \( r_{nk} > H_{nk} \) and 0 otherwise.

4.2. Boolean Networks-Based Consensus. The consensus algorithm of look-up table about conditional variables is an effective approach to perform the consistent TA, however, this algorithm is an empirical method, which is hard to demonstrate by theoretical proof. In this work, the conditional variables are transferred to Boolean logic variables, which are the elements of the Boolean network. Using the semitensor product theory, the Boolean network can be converted into an algebraic function.

The BNAA consensus is shown in Table 1, which lists the decision rules of \( \text{UAV}_n \) according to received information from \( \text{UAV}_m \). There are three actions the UAV can choose from:

(1) Update: The \( \text{UAV}_n \) updates its information about highest bid list \( H_n \) and winner list \( W_n \).

(2) Reset: The \( \text{UAV}_n \) will reset its \( H_{nk} = 0 \) and \( W_{nk} = \emptyset \).

(3) Drop (default action): When \( \text{UAV}_n \) receives the useless data its information will remain the same.

The Boolean network is the core of the consensus algorithm for BNAA. As shown in Table 2, after the conversion of the conditional variables into logical relation, the Boolean network that can be controlled and analyzed by semitensor product approach is established. According to its own information \( \{W_{nk}, b_{nk}, H_{nk}\} \) and received information \( \{W_{mk}, b_{mk}, H_{mk}\} \) from \( \text{UAV}_m \) about Task \( k \), \( \text{UAV}_n \) chooses one action from the decision set \{Update, Reset, Drop\}. The conversion process of variables is shown in Figure 4.

The dynamics of a Boolean control network is expressed as follows \([22]\):
\[
\begin{align*}
\dot{x}_1(t+1) &= \varphi_1(x_1(t), \ldots, x_w(t), u_1(t), \ldots, u_q(t)), \\
\dot{x}_2(t+1) &= \varphi_2(x_1(t), \ldots, x_w(t), u_1(t), \ldots, u_q(t)), \\
&\vdots \\
\dot{x}_w(t+1) &= \varphi_w(x_1(t), \ldots, x_w(t), u_1(t), \ldots, u_q(t)), \\
\dot{y}_j(t) &= f_j(x_1(t), \ldots, x_w(t), u_1(t), \ldots, u_q(t)), \\
&j = 1, \ldots, p,
\end{align*}
\]
where \( x_i(t) \in \Lambda \) denotes logical variables; \( q_i, i = 1, \ldots, w \) and \( f_j, j = 1, \ldots, p \) are the logical functions; \( u_q(t) \in \Lambda \), \( q = 1, \ldots, v \) means the controls; \( y_{ji}(t) \in \Lambda \) indicates the outputs.

A Boolean network can be briefly defined by the five-tuple as
\[
\Omega = (x, y, u, \varphi, f).
\]
where $\mathbf{x} \in \mathbb{R}^{2^n}$ denotes the state variable vector, $\mathbf{y} \in \mathbb{R}^{2^p}$ is the output vector, $\mathbf{u} \in \mathbb{R}^{2^q}$ denotes the control input vector, $\varphi: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ is the state transition logical function, and $f: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ is the output logical function. The dynamics of Boolean network can be formulated as

$$
\mathbf{x}(t+1) = \varphi(\mathbf{x}(t), \mathbf{u}(t)),
$$

$$
\mathbf{y}(t) = f(\mathbf{x}(t), \mathbf{u}(t)).
$$

By the argument as for Theorem 2, if $\mathbf{x} = \kappa_{q=1}^{w} \mathbf{x}_q$, $\mathbf{u} = \kappa_{q=1}^{p} \mathbf{u}_q$, and $\mathbf{y} = \kappa_{q=1}^{p} \mathbf{y}_q$, then these equations can be converted into the vector-logical expression, which is given by

$$
\mathbf{x}(t+1) = \mathbf{M}_\varphi \mathbf{u}(t) \mathbf{x}(t),
$$

$$
\mathbf{y}(t) = \mathbf{M}_f \mathbf{u}(t) \mathbf{x}(t). \quad (17)
$$

According to the logical relation of the variables in this work, the vector variables can be described as follows:

$$
\mathbf{u}_1(t) = \begin{bmatrix} S_1' - S_1 \end{bmatrix}^T,$$

$$
\mathbf{u}_2(t) = \begin{bmatrix} S_2' - S_2 \end{bmatrix}^T,$$

$$
\mathbf{u}_3(t) = \begin{bmatrix} S_3' - S_3 \end{bmatrix}^T,$$

$$
\mathbf{u}_4(t) = \begin{bmatrix} S_H' - S_H \end{bmatrix}^T,$$

$$
\mathbf{x}_1(t) = \begin{bmatrix} S_1 - S_1 \end{bmatrix}^T,$$

$$
\mathbf{x}_2(t) = \begin{bmatrix} S_2 - S_2 \end{bmatrix}^T,$$

$$
\mathbf{y}(t) = \begin{bmatrix} Q_U - Q_U, Q_D, 1 - Q_D, Q_R, 1 - Q_R \end{bmatrix}^T. \quad (18)
$$

Solving (17), the structure matrixes can be received as

$$
\mathbf{M}_\varphi = \begin{pmatrix}
F9F9 & 0000 & 0000 & 0000 \\
0404 & 4444 & 0404 & 4444 \\
0202 & 2222 & FBFB & 2222 \\
0000 & 9999 & 0000 & 9999
\end{pmatrix}_{\text{hex}},
$$

$$
\mathbf{M}_f = \begin{pmatrix}
\delta [F9F9 & 0000 & F9D9 & 0000] \\
\delta [0606 & 7777 & 0626 & 7777] \\
\delta [0000 & 8888 & 0000 & 8888]
\end{pmatrix}_{\text{hex}}. \quad (19)
$$

In many decision systems, the conditional statements can get very complicated, which makes the algorithm difficult to achieve and maintain. On the other hand, the consensus algorithm based on Boolean network by semitensor product theory has important theoretical significance.

5. Simulation

The CBBA and the extended algorithms have demonstrated its advantage in TA and achieved the convergence to conflict-free assignment [12–16]. In addition, the demonstration for mathematical features of Boolean network using semitensor product theory is addressed in many papers [17, 19–22]. Over the mathematical proof, this paper presents a typical scenario and simulation results.

To verify the efficiency of proposed method, the simulation is tested in MATLAB environment on a PC with Intel Pentium Dual-Core CPU E5800 3.2GHz and 4 GB RAM. In the 10 km x 10 km mission scenario, there are 5 UAVs to perform tasks. All UAVs fly at constant velocity of 200 m/s. Experiments are carried out by randomly setting the locations of tasks. Figure 5 compares the performance of CBBA and BNAA. The $x$-axis shows the number of tasks, while the $y$-axis highlights the objective function of each method. The comparison result demonstrates that BNAA can achieve better assignment than CBBA does. Furthermore, Figure 6 gives a simulation result of BNAA in a specific scenario with 10 tasks.
Table 2: The conversion of the conditional variables into logical relation.

(a)
\[
\begin{array}{ccc}
S_1 & S_2 & W_{mk} \\
1 & 1 & m \\
1 & 0 & n \\
0 & 0 & \varnothing \\
0 & 1 & \text{Otherwise} \\
\end{array}
\]

(b)
\[
\begin{array}{ccc}
S'_1 & S'_1 & W_{mk} \\
1 & 1 & m \\
1 & 0 & n \\
0 & 0 & \varnothing \\
0 & 1 & \text{Otherwise} \\
\end{array}
\]

(c)
\[
\begin{array}{ccc}
S_k & \text{Condition} \\
1 & b_{mk} > b_k \\
0 & \text{Otherwise} \\
\end{array}
\]

(d)
\[
\begin{array}{ccc}
S_H & \text{Condition} \\
1 & H_{mk} > H_k \\
0 & \text{Otherwise} \\
\end{array}
\]

(e)
\[
\begin{array}{ccc}
Q_U & \text{Action} \\
1 & \text{Update} \\
0 & \text{Otherwise} \\
\end{array}
\]

(f)
\[
\begin{array}{ccc}
Q_D & \text{Action} \\
1 & \text{Drop} \\
0 & \text{Otherwise} \\
\end{array}
\]

(g)
\[
\begin{array}{ccc}
Q_R & \text{Action} \\
1 & \text{Reset} \\
0 & \text{Otherwise} \\
\end{array}
\]

For a given 5 UAVs, the performance in terms of the time that one-step communication consumed in simulation by varying the number of tasks is shown in Figure 7. Since the Boolean network only involves logical operations, the average time of a communication of BNAA is reduced by 23% compared to CBBA, which utilizes the look-up table to make decision.

6. Conclusion

This paper presents a new way to implement the consensus algorithm of TA for UAVs, which improves theoretical significance and practical value to the development of decision system based on conditional statements. The vector logical
expression, transferred from Boolean network by semitensor product theory, is a connection between conditional statements and mathematical theory. The reason to extend our work based on CBBA is because it is the original algorithm of this series. To strengthen the theoretical effectiveness of consensus algorithm in TA, the BNAA begins with the typically valuable reference for broaching this subject.

Conflict of Interests

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

Acknowledgments

The authors wish to thank the reviewers, the editor, and Dr. Yanfang Liu for many suggestions that substantially improved the quality of this paper.

References
