IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received June 8, 2020, accepted June 11, 2020, date of publication June 15, 2020, date of current version July 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3002594

Dynamic Tasks Scheduling Model of UAV Cluster Based on Flexible Network Architecture

TING DUAN, WEIPING WANG, TAO WANG, XIAOFAN CHEN, AND XIAOBO LI, (Member, IEEE)

College of Systems Engineering, National University of Defense Technology (NUDT), Changsha 410073, China

Corresponding author: Ting Duan (daunting1994@163.com)

ABSTRACT With the rapid growth of application demands and the real-time change of environmental situations, the defects of the UAV task network in adaptability, flexibility, and resilience are becoming more and more prominent. The current network architecture that the junction of points and lines is fixed cannot dynamically provide capacity requirements in real-time due to the failure nodes encountered in the Unmanned Aerial Vehicle (UAV) task scheduling process. To address this challenging issue, this paper proposes a flexible network architecture supporting dynamic fault-tolerant task scheduling model (DSM-FNA) for the UAV cluster. To be specific this paper resorts to super network theory, combining the management theory of flexible network and resilience network to carry out the organizational calculation on the model, and also draw upon linear transformation function to weight and stratify the capability value according to the ability requirement required by the task. Then, a flexible network architecture dynamic scheduling algorithm (FDSA) is proposed, and the substitution strategy is designed for the failure point, to realize the capability and dynamically adapt to the task. Finally, compared with the classical Max-Min algorithm and other algorithms, it is verified that the FDSA algorithm performs better dynamic adjustment for quick response in case of UAV cluster emergencies.

INDEX TERMS Ability cluster, flexible dynamic scheduling, flexible model, task cluster.

I. INTRODUCTION

The rapid development of economic information globalization, the deepening of the concept of unmanned intelligent operation, the rapid progress of AI / ML technology, the application of unmanned platforms in various fields of land, sea, and air, and the adaptive task architecture of Unmanned Aerial Vehicle (UAV) clusters has become a research hotspot. However, most of the existing research ignores the problem of a node or a subgroup failure during the task execution process. Most of the existing researches on UAV cluster mission architecture are from a macro perspective and lack of details and authenticity. How to consider the adhoc emergency response to unexpected situations during the task execution process is the difficulty of the research on the adaptive task architecture of UAV clusters.

Keus' Netforce reference model, whose mathematical model structure can describe two key characteristics of network well [1]. In the current command and control C2 super network model, network nodes and sub-networks have different functions respectively, and the complex relationships in the C2 network are quantified and described [2]. Super network architecture modeling highlights the interaction between nodes while considering the complex and changeable cluster operation environment, great state uncertainty, and strong time constraints, which is a prerequisite for the success of large-scale unmanned operations of both sides [3]. The unmanned combat of a formation can perform different tasks at the minimum cost, and simultaneously carry out comprehensive offensive and disturbance detection missions in multiple combat fields [4]. Mosaic warfare uses dynamic, coordinated, and highly adaptive composable forces to link together low-cost, low-complexity systems in a variety of ways, creating a Mosaic block-like combat system [5]. What makes Mosaic unique is that when parts and combinations of the system are destroyed by the enemy, they automatically and quickly respond to each other, forming a combat system that, though functionally degraded, can still link to each other and adapt to battlefield situations and operational requirements [6]. In the actual combat process, the situation changes in real-time, and the task will change immediately, instead of being fixed in advance. Architecture is the carrier

The associate editor coordinating the review of this manuscript and approving it for publication was Moayad Aloqaily.

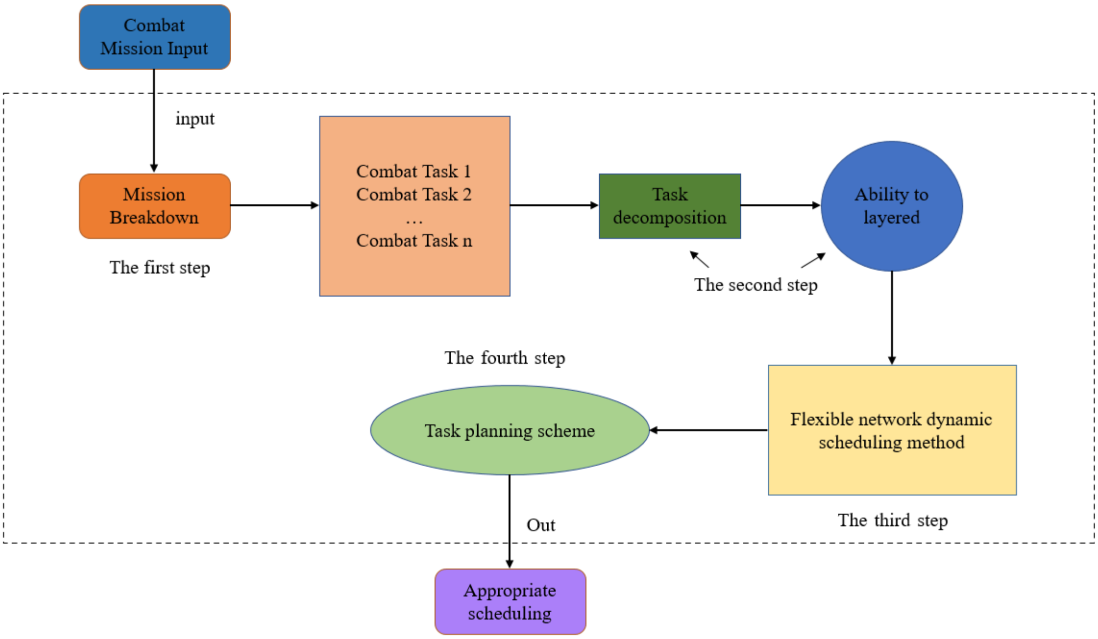


FIGURE 1. The mission planning structure diagram.

of system capability, which integrates all the capabilities of different component systems to achieve the overall mission objectives of the operation. The combat mode has developed from single aircraft simple task to multi-aircraft cooperative execution of multiple complex tasks, and group cooperation, unmanned intervention, and autonomous coordination. Therefore, the integration of the dynamic scheduling model of flexible network architecture (DSM-FNA) into existing combat systems will be the main form of future combat battlefield and play a pivotal role in the development of DSM-FNA capabilities.

In this paper, we advance the state-of-the-art technology in the following ways:

- *First, our team is the first to propose and solve the task of implementing a special event in a UAV cluster architecture, which leads to the failure of the UAV node, and the rapid dynamic reconstruction of the cluster to implement the internal scheduling problem.*
- *Second, a dynamic scheduling model of flexible network architecture for UAV cluster tasks (DSM-FNA) is proposed for the challenges of the current network architecture of the UAV cluster system. The model describes the network architecture by using the theoretical method of a super network to improve the quantitative computing capability of the network.*
- *Third, we use the fuzzy mapping between task node and capability weighted hierarchical method improves the flexibility and intelligence of network architecture and also realizes the mapping relationship between capability, task, and requirement. The cost of searching for space and time is reduced, and the complexity of task scheduling is reduced.*

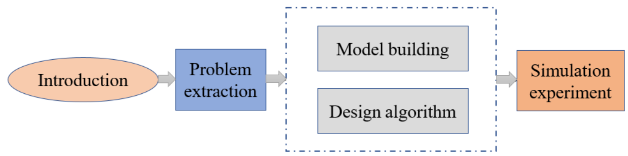


FIGURE 2. Paper organization framework.

- *Fourth, we a flexible dynamic scheduling algorithm (FDSA) for network architecture. To implement the scheduling model we proposed, and compare it with several traditional algorithms, the simulation experiment proves that our research is intentional, and it can be used for the follow-up work of the dynamic scheduling of the UAV cluster architecture.*

The rest of the paper is organized as follows in Fig. 2. In the second part, we introduce the organizational structure of the mission-oriented flexible network architecture of UAV. In the third part, the dynamic scheduling model of DSM-FNA is established to map capability and task requirements effectively. In the fourth part, a flexible dynamic scheduling algorithm (FDSA) for network architecture is designed and compared with the classical Max-Min algorithm and other algorithms. In the fifth part, the FDSA algorithm and classical Max-Min algorithm, and other algorithms are compared and analyzed, and the results are obtained. The sixth part is the research summary of this paper.

II. RELATED WORK

Due to its low cost and diversified combat tasks, unmanned clusters have been attached importance to by military powers, and task scheduling of unmanned cluster architecture has become a research hotspot. In recent years, the United States has been vigorously developing the ''UAV cooperative warfare'' technology, including the new concept of unmanned swarm warfare [11], which is mainly promoted by DARPA, and the loyal wingman project, which is the intelligent technology to control the unmanned ''wingman'' through the fifth-generation fighter aircraft of the United States [12]. All these technologies focus on improving the status and function of intelligent equipment such as large and high-performance UAV, unmanned wingman, and UAV cluster in combat. At present, the research on combat mission scheduling based on the UAV cluster is mainly divided into two parts.

One is to combine task scheduling with an intelligent algorithm to solve the task scheduling problem of UAV cluster architecture. Lamont developed a parallel task planning system based on a multi-objective evolutionary algorithm [13]. Ramirez-Atencia proposed a new multi-objective genetic algorithm to solve the complex task scheduling problem involving a group of unmanned aerial vehicles and a group of ground control stations (GCS) [14]. They further proposed a new algorithm to obtain the most important solution in the Pareto optimal boundary (POF) [15]. Also, due to the complexity of task planning, it is difficult for mathematical models to describe all the characteristics of the planning problem in detail. Because of the complexity of the functional entities of military combat systems and the flow of information between them, modeling them is a challenging task. The rapid development of network science brings new hope. Much research has been done on complex network representations of military organizations [16]–[18].

The second is to use the simulation method to study this problem. Slear [19] designed and implemented a comprehensive mission planning system that integrated several problem domains in the UAV swarm simulation system. Wei *et al.* proposed the operational simulation framework of UAV group configuration and task planning and scheduling [4]. Gaudiano *et al.* [20] proposed a strategy to use a genetic algorithm to evolve group control parameters, such as transfer probability of UAV in different modes, pheromone attenuation rate, and pheromone attraction parameters of UAV, etc., for searching and suppressing enemy air defense missions. Similarly, in Dasgupta [21], the clustering mechanism of automatic target recognition is based on the positive intensified communication mechanism that insects use pheromones to find target paths. The simulation results are only verified on the AEDGE simulation platform. Kurdi *et al.* [22] based on the nature of locust species and their autonomous and resilient behavior to internal and external forces. A new autonomous bio heuristic method is proposed to efficiently allocate tasks among multiple UAVs in one task. The dynamic task allocation mechanism among multiple unmanned aerial vehicles operating autonomously during the mission is discussed [23]. Task assignment is dynamically adjusted by each UAV in the course of the task according to the criteria related to the operational status or mission parameters of the individual UAV.

However, most of the existing studies do not consider the adaptability of the cluster architecture. When the task is performed, the tasks assigned to the UAV are known. During the execution of a task, the real-time change of the situation environment cannot be predicted for everything, failing a certain UAV or a certain subgroup of tasks, and the failure of nodes or subnetworks in the cluster architecture. Ahmad *et al.* used the K shortest path to calculate the optimal path planning path to reflect dynamics [24]. Literature [25] proposes the dynamic resource allocation problem with improved Quality-of-Service applicable to buses. Long-Short Term Memory (LSTM) based neural networks are considered to predict city buses locations for interference determination between moving small cells. It will be a big problem on how to dynamically adapt the unattended cluster architecture to the changing environment.

Valerio *et al.* proposed that flexibility is the ability to recognize emergent behaviors and respond to them [26]. Mandelbaum *et al.* [27] proposed an ability to respond to changes in the environment promptly. Scholars say different things about flexibility. Although there are many arguments about flexible networks, most of them are about ''variability'' and ''invariance''. In this paper, a task-oriented flexible network is proposed given the possibility of node interruption in UAV cluster architecture and the uncertainties encountered at the task end, as well as the disadvantages of network architecture in the aspects of flexibility, adaptability, and self-adaptability. The flexible network model is to use the flexible network theory to serve the operational mission network architecture. The flexible network is to use the resources of the internal network nodes of the current architecture and the effective resources of other nodes in the whole network to meet the requirements of rapid adaptation of the failed nodes and realize the ability to adapt to the sudden changes in the current environment.

III. FLEXIBLE NETWORK ORGANIZATIONAL STRUCTURE

The unattended cluster architecture reflects the configuration of the components in the system and their interaction with the external environment [28], [29]. In the third part, a task-oriented flexible network is proposed given the possibility of node interruption in UAV cluster architecture and the uncertainties encountered at the task end, as well as the disadvantages of network architecture in terms of flexibility and self-adaptability. Using the network of soft change theory of the network, the flexible network is based on the resources of the current structure Intranet node and the effective resources of other nodes in the entire network to meet the rapid change of the failure node, and the ability to adapt to the current sudden environmental change. The flexible network is applied to the adaptive mechanism, and when the situation of the network changes, the shorter time can be adjusted autonomously, the measures can be found in time, and the loss can be reduced as much as possible [30]–[32]. To reduce the scale of the problem, literature [33] proposed a cost method to try to combine the available processing cores into a group

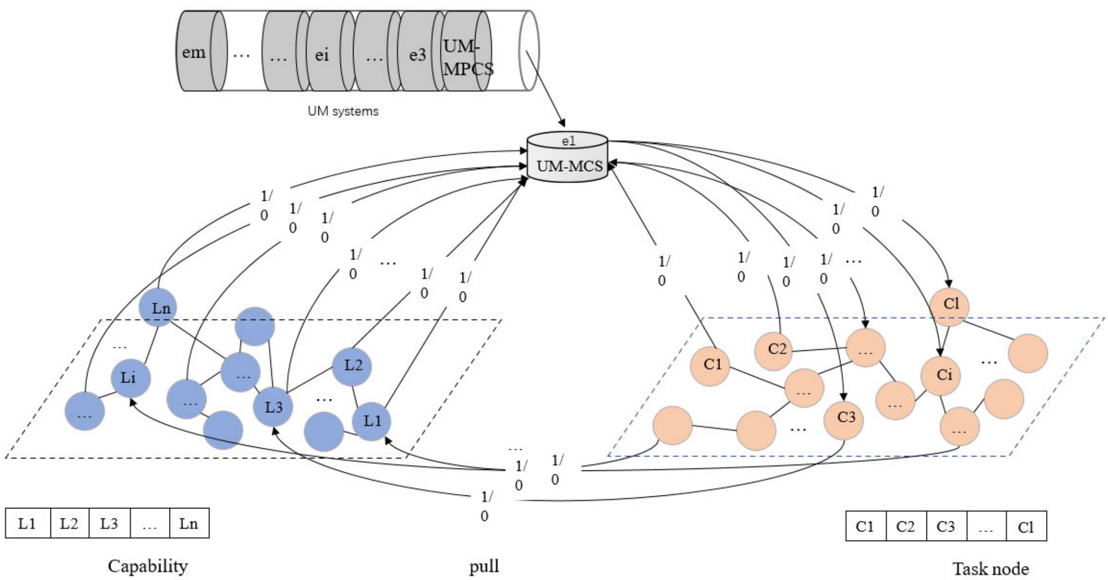


FIGURE 3. UAV cluster mission combat model diagram.

of clusters and allocate the disjoint subset of a given task set. In this aspect, we divide the ability of the task to ability is divided into different capacity levels, and the ability to cluster in the hierarchy is different, to form the ability and the mapping mechanism of the task. Finally, depending on the requirements of the task, select different capacity clusters. We only consider the mutual assistance between drones in the drone cluster and automatically rescue activities. The supply layer is clustered with the desired task requirements.

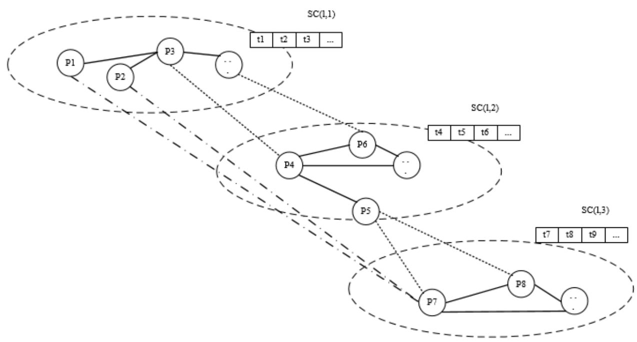


FIGURE 4. Schematic diagram of capability stratification.

A. PROBLEM FRAMEWORK

Definition (Ability): The capability of the combat architecture is that the combat system can provide m capabilities such as reconnaissance and early warning, command and control, and fire interception for the combat system. Let the ability be indexed by l, so $l \epsilon L$, $l = 1, 2, \ldots, m$.

Definition (Ability Cluster): Ability to layer $LC(l, a)$, $l \epsilon [1, SL_n]$, $a \epsilon L = 1, 2, \ldots, m$. Where l is the number of layers, a is the sequence number of power clusters, SC_n is the number of power clusters of clustering, and is denoted $SC(1, 1), SC(1, 2), SC(1, 3), \ldots, SC(n, n)$. The first three are illustrated with a cloud dotted line as shown in Fig. 4 below. By constructing the mapping relationship between the capability supply layer and the task demand, the search space of a single rescue node is reduced to search the entire capability cluster.

Definition (Task Node): A task node is a combat activity that can be towed by a capability node, denoted as C. Our goal is to clearly define the relationship between tasks and capabilities: each task can be broken down into the smallest atomic tasks to form a task list. And can be indexed by n, $n = 1, 2, \ldots, k$. So we build an adjacency matrix for task-capability $C = [C_{ln}]_{mk}$ and about task-capability that says whether task l needs capability n or not. If the task requires an index of capabilities, and set C_{ln} to 1, Otherwise set to 0. For each task n, there is at least one ability l, namely:

$$\sum_{n \epsilon N} C_{ln} \geq 1 (\forall l \epsilon L, C_{ln} = 1 \parallel C_{ln} = 0). \qquad (1)$$

Definition (Task Clusters): Is defined as $TC = t1, t2, t3, \ldots$ It can allocate supplies for tasks, where $t1, t2, t3, \ldots$ as a task queue $p1, p2, p3, \ldots$ is a capability node that provides services. For the task nodes in the network, the flexible network will dynamically send them to the task queue $t1, t2, t3, \ldots$ corresponding to the supply cluster according to the capability of the demand. Defines the set of capability nodes as $P = p1, p2, p3, \ldots, pn$. In order to better describe the capabilities, the three capabilities of UAV, namely, reconnaissance, strike and detection, control and strike, are defined as $p = p_{zc}, p_{cd}, p_{zkd}, \ldots$. Then, the attribute value of the UAV node is calculated numerically to obtain the capability supply layer where the supply node p_i is located.

Definition (System Node): The system node is the equipment to complete the required task and provide the corresponding capability, denoted as D. For the UAS, we set up

TABLE 1. Symbol annotations.

Symbol	Quantity	Symbol	Quantity
L	Ability	LC	ability to layer
SC	the number of cluster	TC	task clusters
SL	the fuzzy capability layer	ZX	the ability cluster center value
$A = [a_{le}]_{mj}$	task-system matrix	D_{enjk}	system-capability matrix
B	the element adjacency matrix	C_{lnmk}	task-capability
σ	the variance	P	capability matrix
n	task node	k	the calculated coefficient
x, y	untransformed property value	μ	the standard deviation
Max	maximum value	Min	minimum value
$M(U, V)$	the objective function		

the UAS for the mission unmanned system (UM-MCS) and the UM combat readiness system as the backup (UM-MPCS). It has j_1 UM execution system and j_1 UM preparation system. Let m_1 be indexed by a_1, it can be expressed as, m_2 be indexed by e_2, it can be expressed as: $e_1 \epsilon E_1, E_1 = 1, 2, \ldots, j_1$. So, the selected system provides the corresponding capability for the task to be performed, so we define the system matrix as: $e_2 \epsilon E_2, E_2 = 1, 2, \ldots, j_2$, m_3 be indexed by e_3, it can be expressed as $e_3 \epsilon E_3 = 1, 2, \ldots, j_3$, so has: $E = E_1 + E_2 + E_3$. The selected system provides the corresponding capability for the task to be performed, so we define the system-capability matrix as $D = [d_{en}]_{jk}$. When system e needs n capabilities, set d_{en} to 1, otherwise it's 0. The task-system matrix is $A = [a_{le}]_{mj}$. When task l needs e systems, set a_{le} to 1, otherwise set it to 0. Both:

$$\sum_{n \epsilon N} d_{en} \geq 1(\forall e \epsilon E, d_{en} = 1 \parallel d_{en} = 0). \tag{2}$$

$$\sum_{e \epsilon E} a_{le} \geq 1(\forall l \epsilon L, a_{le} = 1 \parallel a_{le} = 0). \tag{3}$$

where, we assume that each system will provide more than one capability, that each task activity selects more than one system, and that the different capabilities satisfy the linear sum.

Definition (C2 Organization): We assume that MCSs is controlled by $u_1 C_2$ cell and all cells constitute u_2 population, and C_2 cell and population are pulled by $s_1 s_2$, that is,

$$\begin{cases} S_1 = [1, 2, \ldots, u_1], & \forall s_1 \epsilon S_1, \\ S_2 = [u_1 + 1, u_1 + 2, \ldots, u_1 + u_2], & \forall s_2 \epsilon S_2, \\ S_3 = [u_2 + 1, u_2 + 2, \ldots, u_3 + u_2], & \forall s_3 \epsilon S_3, \\ S = S_1 + S_2 + S_3. \end{cases} \tag{4}$$

At the same time, each C2 unit in the UM combat system was pulled by s, and each group was assigned to a C2 unit. Finally, the cluster was integrated into the UM combat system and UM combat preparation system, respectively forming the adjacency matrix B, C, and D. In the reconnaissance system, if task e_1 is assigned to s_1 in C2 cell, set $b_{e1s1} = 1$ otherwise be 0, then constitute the element adjacency matrix $B = [b_{e1s1}]_{j1u1}$, the element adjacency matrix of the detection system is $C = [c_{e2s2}]_{j2u2}$. Similarity, the element adjacency matrix of the detection and control system is $D = [d_{e3s3}]_{j3u3}$. With so many symbols and notations, we introduce a table that is easy to find such as Table 1.

B. ABILITY TO LAYERED

In this article, to reduce the size of the problem, we only consider dividing the inherent capabilities into layers. In addition, this paper only considers the three inherent attributes of UAV's capability resources: reconnaissance capability, attack capability, and detection and control capability. Next, the inherent capabilities are divided into several layers, and the set of capabilities is defined as $P = P_1, P_2, \ldots, P_N$. Its attribute is $p = p_{zc}, p_{cd}, p_{zkd}, \ldots$, and the three main research objects are mainly discussed, namely, the reconnaissance capability p_{zc}, the strike capability p_{cd}, and the detection, control and attack capability p_{zkd}. Then the data is preprocessed to form the capability layer of different capability performance levels $SL_l(l \epsilon [1, SL_n])$.

The method of linear transformation function and standard deviation are used to normalize the attribute value of capability. Firstly, the linear transformation function is used to normalize the capability inherent attributes of network architecture nodes. In order to facilitate calculation and programming, the converted inherent attribute values are divided into (0,1). The conversion function is expressed as follows:

$$k = (b - a)/(Max - Min) \tag{5}$$

$$y = b + k * (x - Max) \tag{6}$$

Here k is the calculated coefficient, and a and b are the intervals, and $a = 0$, $b = 1$. The expressions of (5)-(6). are used to normalize the natural properties of the supply layer, where x is the untransformed property value, y is the transformed property value, and Max and Min respectively represent the maximum and minimum values of the inherent properties in the function. The membership function in fuzzy mathematics is used to divide the ability of the supply layer into the following levels:

$$SL_1(x) = \begin{cases} 0 & x > 0.2 \\ (0.2 - x) * 10 & 0.1 \le x \le 0.2 \\ 1 & 0 \le x < 0.1 \\ 0 & x < 0 \end{cases} \tag{7}$$

$$SL_2(x) = \begin{cases} 0 & x > 0.4 \\ (0.4 - x) * 10 & 0.3 \le x \le 0.4 \\ 1 & 0.2 \le x < 0.3 \\ (x - 0.1) * 10 & 0.1 \le x < 0.2 \\ 0 & x < 0.1 \end{cases} \tag{8}$$

$$SL_3(x) = \begin{cases} 0 & x > 0.6 \\ (0.6 - x) * 10 & 0.5 \le x \le 0.6 \\ 1 & 0.4 \le x < 0.5 \\ (x - 0.3) * 10 & 0.3 \le x < 0.4 \\ 0 & x < 0.3 \end{cases} \tag{9}$$

$$SL_4(x) = \begin{cases} 0 & x > 0.8 \\ (0.8 - x) * 10 & 0.7 \le x \le 0.8 \\ 1 & 0.6 \le x < 0.7 \\ (x - 0.5) * 10 & 0.5 \le x < 0.6 \\ 0 & x < 0.5 \end{cases} \tag{10}$$

$$SL_5(x) = \begin{cases} 0 & x > 1 \\ 1 & 0.8 \le x < 1 \\ (x - 0.7) * 10 & 0.7 \le x < 0.8 \\ 0 & x < 0.7 \end{cases} \tag{11}$$

To better represent, this paper visualizes the membership function of the MATLAB capability layer in the following figure, where are the different capability layers of the network architecture capability respectively, forming a collection of the fuzzy layer. $SL_1, SL_2, SL_3, SL_4, SL_5$ is a set of different capability layers of network architecture capabilities, and its corresponding membership functions can be expressed as expressions (7)-(11), and then normalized by using equations (5)-(6).

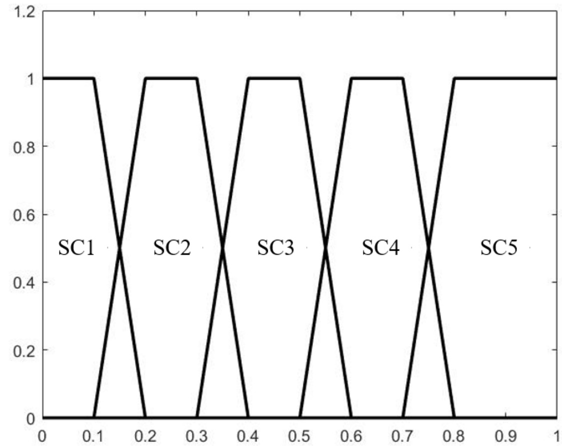


FIGURE 5. **The membership of the capability layer.**

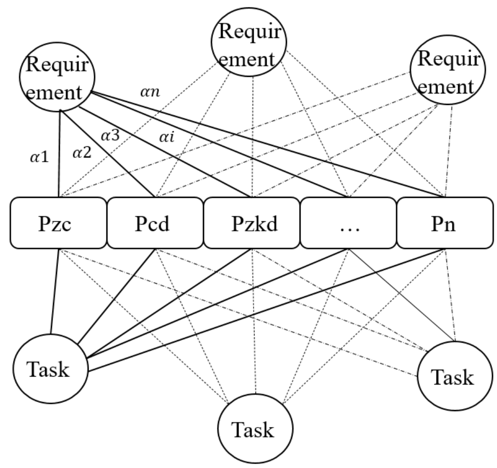


FIGURE 6. **Capacity requirements and task mapping.**

C. ABILITY IS QUANTIFIABLE

The maximum value is set randomly at the beginning. The three inherent attributes are normalized as p_{zc}_n, p_{cd}_n, and p_{zkd}_n by using equation (5), and then the values of the three inherent attributes are normalized by using the calculation expression (12)-(13), which can be expressed as $\overline{p_{zc}}, \overline{p_{cd}}, \overline{p_{zkd}}$.

$$\mu = [p_{zkd}\_n + p_{zc}\_n + p_{cd}\_n]/3. \tag{12}$$

$$\sigma = ([(p_{zkd}\_n - \mu)^2 + (p_{zc}\_n - \mu)^2 + (p_{cd}\_n - \mu)^2]/3)^1/2. \tag{13}$$

$$\overline{p_i} - score = (p_i - \mu)/\sigma. \tag{14}$$

where μ represents the standard deviation, σ represents the variance, and formula (2) represents the normalization of the values of the three inherent attributes of p_{zc}_n, p_{cd}_n and p_{zkd}_n. Then, the membership degree of each ability can be calculated by substituting the membership degree function into three attribute values respectively, which are expressed as $SC_L(\overline{p_{zc}})$, $SC_L(\overline{p_{cd}})$, $SC_L(\overline{p_{zkd}})$, where $L\epsilon[1, SC_n]$. The membership degree of each corresponding capability layer is defined as:

$$SC_L(P_N) = d * SC_L(\overline{p_{zc}}) + e * SC_L(\overline{p_{cd}}) + f * SC_L(\overline{p_{zkd}}). \tag{15}$$

The evaluation expression (5) represents the membership of P_N to the ability layer SC_L, and the article uses the d, e, f to represent the weight value of each attribute value, and the weight is in line with the demand for the current loss of the UAV node. When an unmanned vehicle fails to complete the task, the task that it needs to do is to require a higher capacity for reconnaissance, and the lower requirements for other capabilities, which can make d bigger, e and f smaller and make $d + e + f = 1$. After layering, the maximum membership is selected in order from large too small to obtain the ability layer $Max[SC_L(N_N)]$ of $Max(N_N)$.

D. CAPACITY CLUSTER CALCULATION

Consider the multi-node failure case, which means that multiple nodes have requirements. As the number of requirements nodes increases, there are capabilities of the same type for each requirement that perform the same tasks. We set up different capability layers that can provide the same requirement purpose. Each capability layer contains multiple capability clusters, and each capability cluster can provide similar requirements and perform similar tasks. In this paper, the linear transformation function is used to normalize and standardize data preprocessing, and the clustering into different capability clusters is denoted as $SC(l, a)$.

The ability set in the L layer is called $P_N = p_1, p_2, \ldots, p_i, \ldots, p_N$, and p_i is the first i capability subset in the capacity set. The center set of the ability cluster is defined as $Z = z_1, z_2, \ldots, z_k, \ldots, z_m, j\epsilon 1, m$. The calculation expression of capability cluster center z_j is denoted as:

$$ZX = \sum_{i=1}^{n}(w_{ij} * p_i)/\sum_{j=1}^{m} w_{ij}. \tag{16}$$

$$M(U, V) = \sum_{j=1}^{m}\sum_{i=1}^{n}(w_{ij}^2) * [\sum_{i=1}^{n}(p_i, z_j)]^2. \tag{17}$$

The objective function depends on the membership function, and the clustering of the supplying capability depends on the central minimum objective function of the capability cluster.

$$\begin{aligned} w_{ij} &= dist(p_i, ZX_j)/dist(p_i, ZX_j). \\ &= \sqrt{\sum_{i=1}^{n}}(p_i - ZX_i)^2/\sqrt{\sum_{i=1}^{n}}(p_i - ZX_j)^2 \end{aligned} \tag{18}$$

The membership degree of the ability cluster is obtained by using the membership degree function to form a set. To solve the objective function according to the calculation expression (18), each ability will be divided into the set of the ability cluster with the highest membership degree.

The summary process is as follows: firstly, the initial sample matrix is established by the requirements provided by m capabilities, and the data is normalized by using equations (5)-(6), and then the data set is standardized by using equations (14). Then the number of capability clusters is determined and the membership matrix is obtained by using the objective function $M(U, V)$. Then, according to the calculation expression (18), the ZX of the ability cluster center is calculated. Then, according to the demand, the objective function is used to calculate the value of the objective function of the ability cluster center, and the membership matrix is updated until the clustering requirements are met.

E. CAPABILITY REQUIREMENTS TASK MAPPING

Capabilities are mapped to requirements in combat activities by taking into account the capabilities of the three inherent attributes p_{zc}, p_{cd} and p_{zkd}. The mathematical expectation of discrete random variables is used for calculation. The capability values of inherent attributes are first comprehensively considered and then the corresponding capability clusters are determined according to the two-dimensional relation of demand capability, to realize the mapping of demand and capability and meet the requirements of different combat tasks.

When a requirement instruction is issued, the weight of capability value tends to be different according to different requirements. Similarly, the correspondence relation of task capability value is considered comprehensively. As shown in Fig. 5. So, using the expected value,

$$E(P_i) = \sum_{i=1}^{n}(\partial_i * p_i) \tag{19}$$

In this paper, to simplify the calculation and reduce the complexity of the problem, only p_{zc}, p_{cd} and p_{zkd} are considered. That is:

$$BS(P) = \partial_1 * p_{zc} + \partial_2 * p_{cd}\,\partial_3 * p_{zkd} \tag{20}$$

where, parameter $\partial_1, \partial_2, \partial_3$ represents the empirical coefficients of reconnaissance capability, control capability, and detection and control capability respectively, and the sum of the three empirical coefficients is 1. According to the calculation expression (19), it can be expressed as(20), Where, p_{zcx}, p_{cdx} and p_{zkdx} are respectively the requirements for the capability values of p_{zc}, p_{cd} and p_{zkd}, when the task needs to be executed, and $\beta_1\beta_2\beta_3$ is the weight of the calculation expression (21), and the sum is 1.

$$XN(P) = \beta_1 * p_{zcx} + \beta_2 * p_{cdx}\beta_3 * p_{zkdx} \tag{21}$$

IV. FLEXIBLE DYNAMIC SCHEDULING ALGORITHMS (FDSA)

A. SCHEDULING ALGORITHM DESIGN

Next, this paper presents the dynamic scheduling algorithm flow and its pseudo-code of the UAV task architecture network based on the flexible network architecture, which is automatically started when confronted with the emergent situation.

The flow of the algorithm is as follows.

(1) Send task requirements to the corresponding capability cluster. When a task monitoring robot finds the existence of a task request, according to the requirements of the task, evaluation expression (12) is used to calculate the comprehensive capability expectation, find the corresponding capability cluster, and then the robot sends the information of the task to the virtual task queue.

(2) The alternate strategy of starting ability. The generates the task is used as a signal to automatically trigger the alternate strategy, and the co-managed robot begins to wait for the evaluation and co-load value from other abilities.

(3) Calculate the residual capacity and workload, and judge whether to participate in the substitution strategy. The ability of cluster in its capacity surplus value can be calculated respectively and according to the following expression (18) own synergy load network architecture, to the current task of virtual task queue in the group of the ability to read the

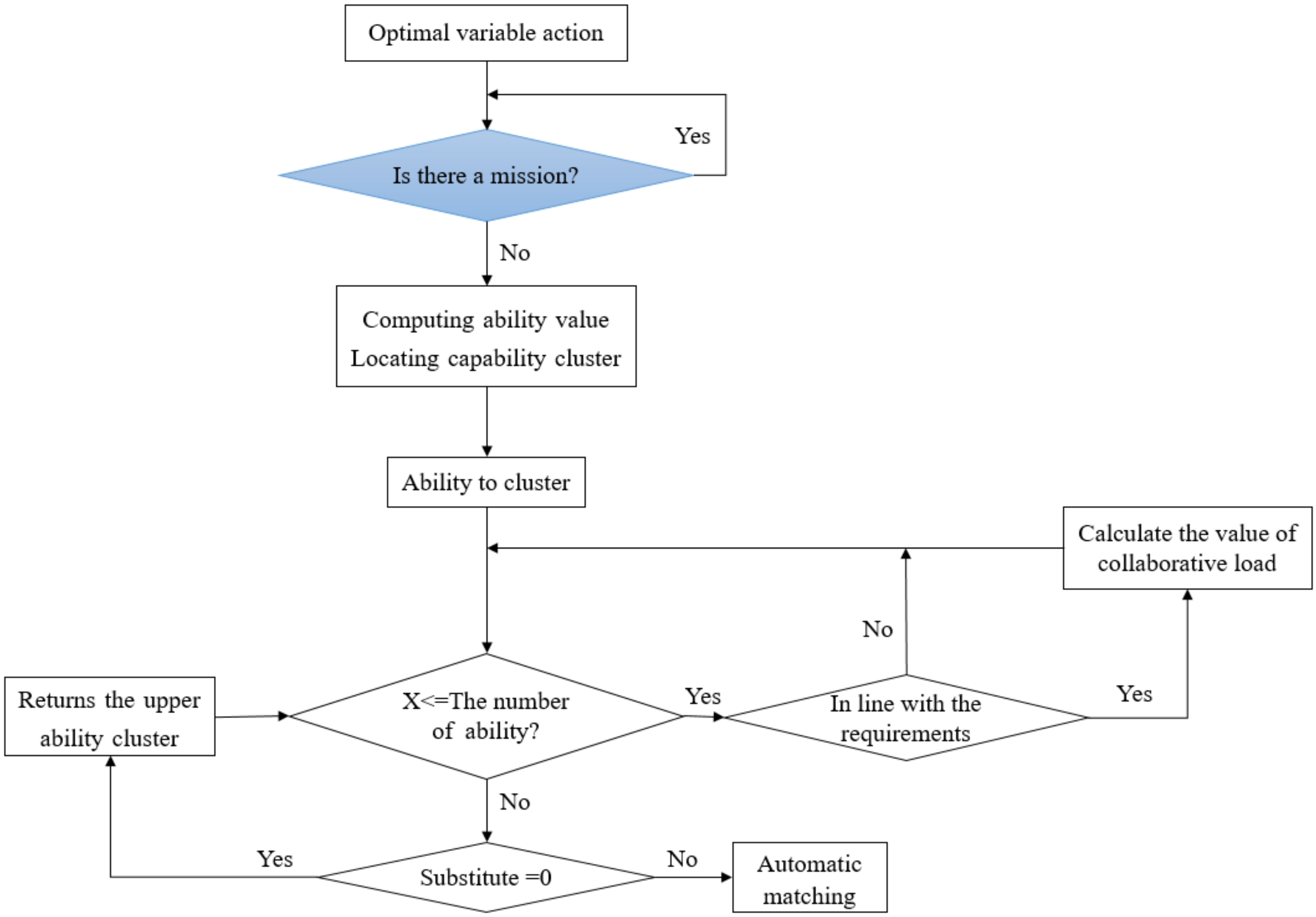


FIGURE 7. FDSA algorithm scheduling diagram.

information. If you meet the requirements, participate in the reserve. If unable to meet, exit the current layer from the upper level of the ability cluster.

(4) Calculate the evaluation value and collaborative load value of the task, and then send the evaluation result through the information interaction robot.

(5) Determine the target node of the execution task. The robot automatically sets the execution point of the task node with the maximum evaluation value to the capability node that provides the maximum evaluation value.

(6) Perform tasks. When the backup strategy ends, i.e., the backup is zero, the current UAV is automatically replaced, the mission is executed, and the operation is finished.

B. NETWORK ARCHITECTURE SCHEDULING METHOD PROCESS

In the flexible network computing model, capabilities are divided into clusters of different capability layers and can be represented by different sets. Every time a task meets a special situation, it is assigned according to the need, and the task needs to find the substitute in the corresponding ability cluster according to the demand of the ability, thus reducing the completion time of the task. As shown in the following matrix.

$$\begin{bmatrix} SC(1,1) & SC(1,2) & \cdots & SC(1,i) & \cdots & SC(1,SC_n) \\ SC(2,1) & SC(2,2) & \cdots & SC(2,i) & \cdots & SC(2,SC_n) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ SC(SL_n,1) & SC(SL_n,2) & \cdots & SC(SL_n,i) & \cdots & SC(SL_n,SC_n) \end{bmatrix} \tag{22}$$

When a task is sent to the corresponding capability cluster, a new task will only send one message to generate one load value, and m load values will be generated for m tasks. When the virtual task queue is not empty, each team member who can participate in the substitution needs to read information to determine whether he can join the substitution queue. At this time, the maximum load value generated is n pieces of information acceptable to the capability node that can perform the task. When the scheduling policy is started to select a substitute, the value of the information generated in the whole process will not be greater than the m message sent by the task. All UAV cluster alternate nodes identify the UAV capability node with the highest evaluation value as the alternate node of the failure node and then send the task to the alternate UAV node. Therefore, the load value generated for m tasks is no more than m. That is, the total load value

Algorithm 1 Scheduling Algorithm FDSA

Input: initialize a string T at random;
Output: until(termination-condition)
while Task request(TR)= null **do**
  Calculate the combined expectations
  determine the capacity cluster, and send the task
  while $x <= m$ **do**
    Calculate the residual capacity (RC) and the collaborative load values (CLV)
    Read the task information of the virtual task queue
    if ($RC >= TR\&CLV <= Thethresholdvalue$) **then**
      Computational ability evaluation
      The robot conveys the value
    end if
  end while
  if ($BENCH == 0$) **then**
    Calculate the maximum value of evaluation & set provide node Automatic matching & performing tasks
  else
    while $l < SC_L$ **do**
      Search for search from the previous layer
      Returns while(x<=m)
      BREAK
    end while
  end if
end while

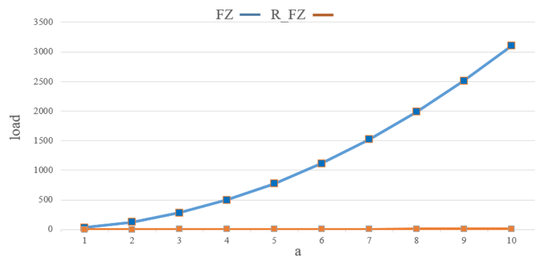


FIGURE 8. Contrast with traditional random loads.

generated by the comprehensive calculation is:

$$R\_FZ = m + n + m + m = n + 3m. \quad (23)$$

The next calculation is the increasing of the network architecture, the ability node n, and the task node m. As the scale of the problem increases, the relationship between the ability of the attribute and the requirements of the task is not only a great reduction in the complexity of the problem, but also improves the efficiency of the task. When the failed node and the unexpected situation occur, the suitable task replacement points can be quickly found in the capacity cluster produced by the hierarchical subhierarchy. To compare random loads, we obtained Fig. 8 based on the data Table 2.

V. EXPERIMENTAL VERIFICATION

To verify that DSM-FNA can effectively adapt UAV tasks and capability requirements, reduce scheduling time and system communication load, we implemented the algorithm using Java HotSpot(TM) 64-bit Server VM on the eclipse platform in Win64 Graphics Environment and compared it with the traditional task scheduling method max-min under the same parameter setting.

TABLE 2. Contrast with traditional random loads.

Order a	$m = 10n$	$n = a^2$	FZ	R_FZ
1	10	1	19	31
2	40	4	319	124
3	90	9	1619	279
4	160	16	5119	496
5	250	25	12499	775
6	360	36	25919	1116
7	490	49	48019	1519
8	640	64	81919	1984
9	810	81	131219	2511
10	1000	100	199999	3100

A. PARAMETER PERFORMANCE SETTINGS

In the experimental environment, the ability parameter value and the number of tasks were given in advance. The capacity storage space is calculated using the computer's storage range of [480mb,480gbds], and the Java version is $1.8.0\_202 - ea$. In order to evaluate the analysis, we use the scheduling completion time and the C2 organization's collaborative load CW to experiment with the comparison. Scheduling completion time depends on the completion time of the virtual task queue,

$$T_{zx} = \sum_{e1=1}^{j1} b_{e1s1} + \sum_{e2=j1+1}^{j1+j2} f_{e2s2}. \quad (24)$$

The completion time $Time$ depends on the time Ttx and the task execution time of Tzx.

$$Time = Ttx + Tzx = \sum_{l=1}^{N} (Ttx_l/Ptx_l + Tzx_l/Pzx_l). \quad (25)$$

The collaborative load depends on the communication load between the task and the node and can be expressed by the correlation coefficient. The greater the correlation coefficient of the substitute UAV crew is, the stronger the relationship between the two variables will be, and the smaller the cooperative load will be. Therefore, the cooperative load of CW can be expressed by the Pearson correlation coefficient:

$$CW = \rho(P_i, P_j) = \frac{Cov(P_i, P_j)}{\sigma P_i \sigma P_j} = \frac{E[(P_i - \mu P_i)(P_j - \mu P_j)]}{\sigma P_i \sigma P_j}. \quad (26)$$

In this, P_i, P_j belongs to the capability layer $P_N = p_1, p_2, \ldots, p_i, \ldots, p_N$, and the corresponding relationship between the comprehensive consideration of the task capability is calculated in evaluation expression (26).

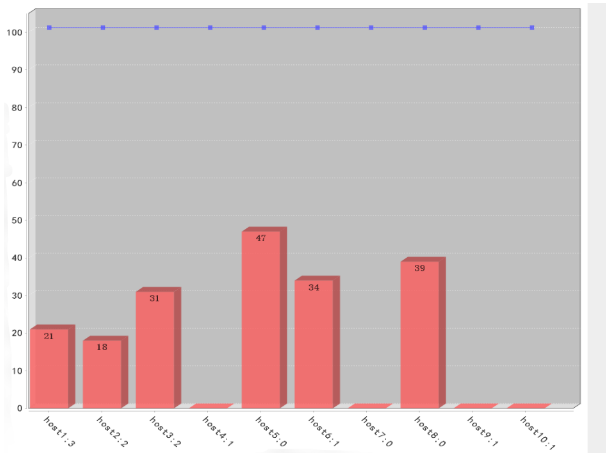


FIGURE 9. Sample figure of FDSA algorithm scheduling 10-6.

B. ANALYSIS OF EXPERIMENTAL RESULTS

To verify that DSM-FNA can effectively reduce the system communication load and task scheduling completion time, this paper uses FDSA scheduling, randomly input 10 unmanned aircraft to perform 6 tasks, and gives the task scheduling diagram in Fig. 9.

The FDSA algorithm was compared with the classical Max-Min algorithm, the classical Min-Min algorithm, and the FDSA variant algorithm $F_T ES$. The simulation experiment is designed as follows: when the task number is the same and the ability resources are different. This experiment observes the communication load of the middle class of the scheduling process.

Randomly set the virtual task queue VSQ = 1330, 1500, 1550, 1650, 1650, 1750, 1750, 1750, 1850, 1850, 1900, 2000, 2050, 2100, 2150, 2200, 2250, 2300, 2350, While the capacity supply number AN = [8000, 10000], when the number of tasks is the same but the number of resources is different, the total task completion time of the two tasks is observed in this experiment. The comparative analysis of experimental results is shown in the following.

(a) The number of task sequences is the same as the capability value at the same time, and the time comparison generated by the scheduling process. Randomly set the virtual task queue VSQ = 1330, 1500, 1550, 1650, 1650, 1750, 1750, 1750, 1850, 1850, 1900, 2000, 2050, 2100, 2150, 2200, 2250, 2300, 2350. While the capacity supply number AN = [8000, 10000], when the number of tasks is the same but the number of resources is different, the total task completion time of both is observed in this experiment. The comparative analysis of experimental results is shown in below, where Fig. 10(a) represents the time comparison between FDSA under the same task sequence and other traditional scheduling algorithms when AN = 8000, and Fig. 10(b) represents the time consumption comparison when AN = 10000.

(b) Time comparison generated by the scheduling process when the number of task sequences is different and the capability value is the same. Randomly set the virtual task queue VSQ1 = 1330, 1500, 1550, 1650,

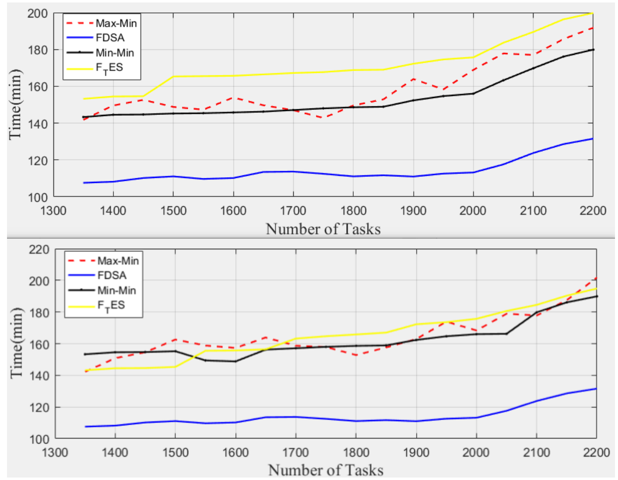


FIGURE 10. Compare the time spent on tasks with the same number of abilities.

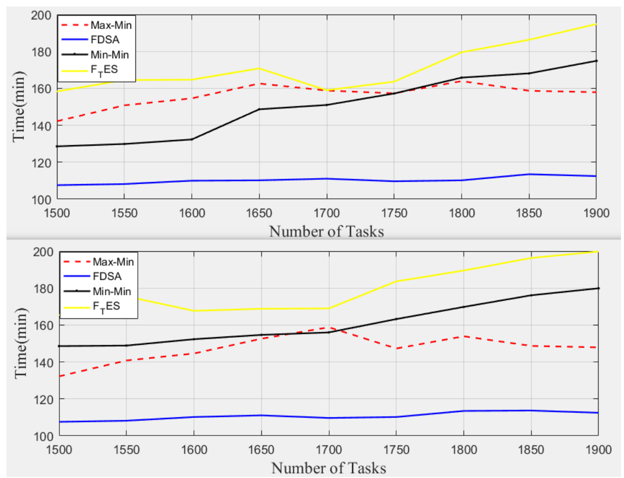


FIGURE 11. Comparison of scheduling time of different tasks with different abilities.

1650, 1650, 1750, 1750, 1800, 1850, VSQ2 = 1900, 2000, 2050, 2100, 2150, 2200, 2250,2 300, 2350. However, the capacity supply number AN = 8000. When the number of tasks is different but the number of abilities is the same, the total completion time of both tasks is observed in this experiment. The comparative analysis of experimental results is shown in Fig. 11, where Fig. 11(a) represents the time comparison between FDSA and traditional scheduling algorithm when VSQ1, and Fig. 11(b) represents the time comparison between FDSA and other traditional scheduling algorithms when VSQ2.

(c) A comparison of loads with the same number of virtual tasks and different Numbers of capabilities. Compare the number of virtual tasks with the same number of abilities. Where, Fig. 12(a) indicates that when AN = 8000, the co-load value CW of FDSA under the unified task sequence is compared with that of the Max-Min scheduling algorithm. Fig. 12(b) indicates that when AN = 10000, the co-load value CW of FDSA under a unified task sequence is compared with that of the Max-Min scheduling algorithm, the Min-Min

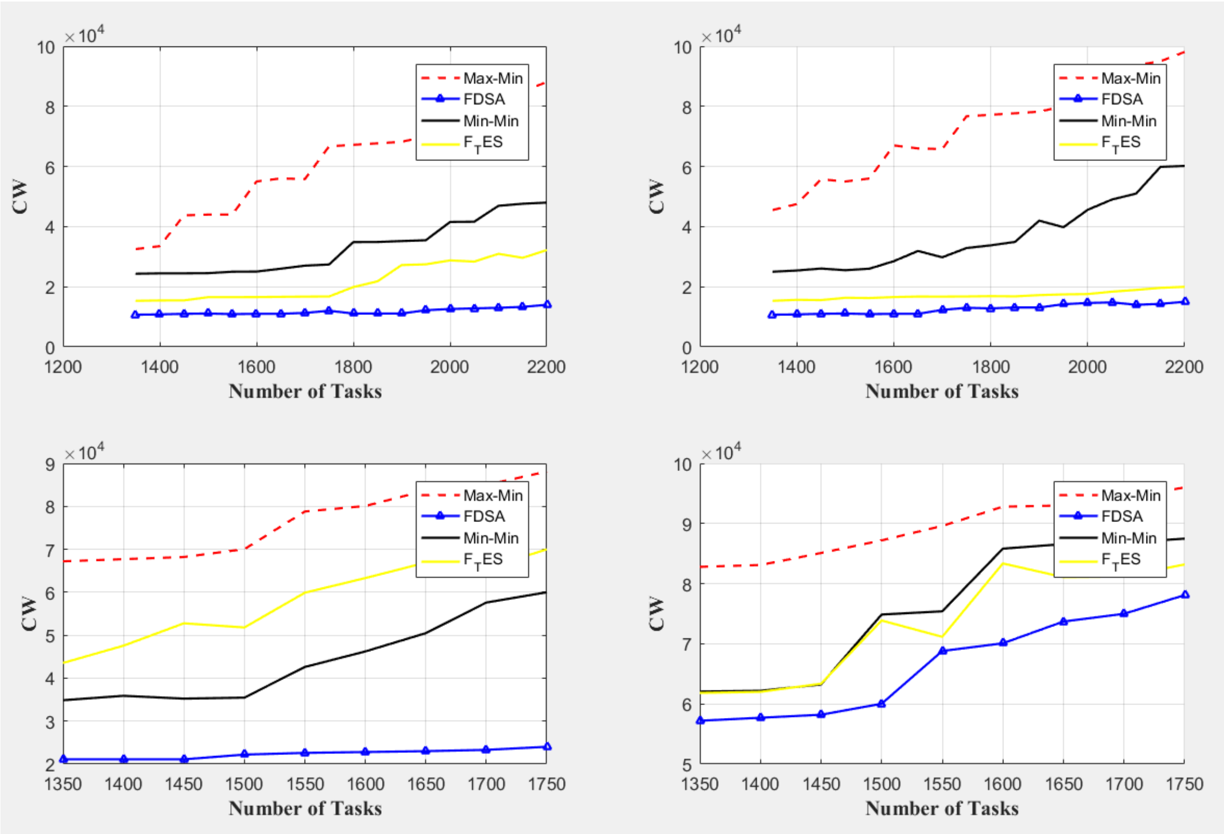


FIGURE 12. Collaborative load value comparison.

scheduling algorithm, and the F_TES algorithm. Fig. 12(c) shows the load comparison between FDSA and several classical scheduling algorithms at VSQ1, and Fig. 12(d) shows the load comparison between FDSA and traditional scheduling algorithms at VSQ2.

As can be seen from Fig. 10-12, the maximum task scheduling of the FDSA algorithm takes less time than that of the other scheduling algorithms. This is because DSM-FNA implements the mapping between capability requirements and tasks. According to the requirement of capability value, the FDSA algorithm determines the node to execute the service. Therefore, the FDSA algorithm embodies the principle of on-demand allocation and effectively shortens the completion time of the task. From the above experiments, it can be seen that DSM-FNA can effectively reduce the maximum task scheduling time and system communication load. This is mainly because the FDSA algorithm considers the mapping between capabilities and task requirements in the scheduling process. According to the comprehensive expectation, the selection can perform the calculation and generate less load value. In general, the replacement strategy process of the FDSA algorithm only includes the ability to meet the task requirements, while the classical scheduling algorithm requires all UAV network capability nodes in the system to participate in the replacement strategy process. At the same time, we analyzed the average difference method log and found that the FDSA improved by percent 40.12. Therefore, FDSA has less traffic load than other scheduling algorithms.

VI. CONCLUSION

Aiming at the challenge of flexibility in the current UAV cluster task scheduling network architecture, a Dynamic scheduling model of flexible network architecture (DSM-FNA) for UAV cluster tasks is proposed and established. DSM-FNA designs an FDSA based on a flexible network to dynamically manage and control capabilities and tasks, effectively realizing the dynamic adaptation of capabilities and tasks. Based on the comprehensive capability expectation, the capability cluster in the capability layer can be corresponding, and the capability and task can be mapped one by one. Therefore, the selection space is effectively reduced, the operational needs of different task scheduling are satisfied, and the rationality of assignment is guaranteed. Finally, through experimental comparison and analysis, it can be concluded that FDSA algorithm can effectively reduce the communication load and task scheduling time compared with the classical Max-Min algorithm and other algorithms, which verifies the effectiveness of the proposed model, guarantees the flexibility and reliability of network architecture, and provides a theoretical basis and key technology for dynamic matching of task requirements and capabilities. Thus, the flexibility, reliability, and independent decision-making ability of the system are improved, and the problem of adaptability of the task architecture in a small range under the change of situation environment is satisfied correctly. The future work will focus on the problem that the failure node is going to select a given number of optimal architectures from the UAV cluster architectural space. Further, we hope to study the problem of failure node capacity in DSM-FNA.

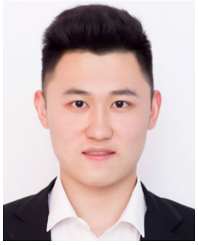
REFERENCES

[1] D. Konur, H. Farhangi, and C. H. Dagli, ''On the flexibility of systems in system of systems architecting,'' *Procedia Comput. Sci.*, vol. 36, pp. 65–71, 2014.
[2] I. Cherfa, N. Belloir, S. Sadou, R. Fleurquin, and D. Bennouar, ''Systems of systems: From mission definition to architecture description,'' *Syst. Eng.*, vol. 22, no. 6, pp. 437–454, Nov. 2019.
[3] B. Chen, H. Yu, Y. Wang, X. Gao, and Y. Xu, ''Multilevel command and control supernetwork modeling based on attribute synergy prioritization,'' *IEEE Access*, vol. 7, pp. 32693–32702, 2019.
[4] Y. Wei, M. B. Blake, and G. R. Madey, ''An operation-time simulation framework for UAV swarm configuration and mission planning,'' *Procedia Comput. Sci.*, vol. 18, pp. 1949–1958, 2013.
[5] L. G. D. Deptula, M. G. L. Stutzriem, H. Penney, and M. Gunzinger, ''Mosaic wars: Restoring America's military competitiveness,'' Mitchell Inst. Aerosp. Stud. Air Force Assoc., Arlington, VA, USA, Tech. Rep., Sep. 2019, pp. 8–53.
[6] B. Clark, D. Patt, and H. Schramm, ''Mosaic warfare exploiting artificial intelligence and autonomous systems to implement decision-centric operations,'' CSBA, Center Strategic Budgetary Assessments, Washington, DC, USA, Tech. Rep., 2020, pp. 27–39.
[7] G. Selvakumar, K. S. Ramesh, S. Chaudhari, and M. Jain, ''Throughput optimization methods for TDMA-based tactical mobile ad hoc networks,'' *Bharat Electron.*, pp. 978–981, 2019.
[8] J.-F. Castet and J. H. Saleh, ''On the concept of survivability, with application to spacecraft and space-based networks,'' *Rel. Eng. Syst. Saf.*, vol. 99, pp. 123–138, Mar. 2012.
[9] X. Gao, H. Yu, Y. Wang, and B. Chen, ''A modeling method for command and control supemetworks based on hyperedge generation strategies,'' in *Proc. IEEE 22nd Int. Conf. Comput. Supported Cooperat. Work Design ((CSCWD))*, May 2018, pp. 128–132.
[10] J. Xu, Y. Yin, X. Wen, and G. Lin, ''Research on the supernetwork equalization model of multilayer attributive regional logistics integration,'' *Discrete Dyn. Nature Soc.*, vol. 3, pp. 1–9, Nov. 2019.
[11] C. Yuan, ''DARPA aviation development summary,'' *Conmilit*, vol. 3, pp. 39–40, 2017.
[12] Y. Qing-Feng and Y. Jian-Ming, ''How to view and respond to drone swarm warfare?'' *PLA Daily*, vol. 7, Aug. 2017.
[13] G. B. Lamont, J. N. Slear, and K. Melendez, ''UAV swarm mission planning and routing using multi-objective evolutionary algorithms,'' in *Proc. IEEE Symp. Comput. Intell. Multi-Criteria Decis.-Making*, Honolulu, HI, USA, Apr. 2007, pp. 10–20.
[14] C. Ramirez-Atencia, G. Bello-Orgaz, M. D. R-Moreno, and D. Camacho, ''Solving complex multi-UAV mission planning problems using multi-objective genetic algorithms,'' *Soft Comput.*, vol. 21, no. 17, pp. 4883–4900, Sep. 2017.
[15] C. Ramirez-Atencia, S. Mostaghim, and D. Camacho, ''A knee point based evolutionary multi-objective optimization for mission planning problems,'' in *Proc. Genetic Evol. Comput. Conf.*, Jul. 2017, pp. 1216–1223.
[16] P. Di, T. Hu, B. Hu, and J. Zhang, ''Research on invulnerability of combat net model based on complex networks,'' *J. Syst. Simul.*, vol. 23, no. 1, pp. 56–60, 2011.
[17] Y. Qi, Z. Liu, H. Chen, and J. Han, ''Distributed combat system of systems network modeling,'' *J. Netw.*, vol. 8, no. 8, pp. 1906–1912, Aug. 2013.
[18] J. R. Cares and J. Q. Dickmann, *Operations Research for Unmanned Systems*. New York, NY, USA: Wiley, Jul. 2016.
[19] J. N. Slear, ''AFIT UAV swarm mission planning and simulation system,'' Ph.D. dissertation, Dept. Elect. Comput. Eng., AFB, El Segundo, CA, USA, 2006.
[20] P. Gaudiano, E. Bonabeau, and B. Shargel, ''Evolving behaviors for a swarm of unmanned air vehicles,'' in *Proc. IEEE Swarm Intell. Symp.*, Jun. 2005, pp. 317–324.
[21] P. Dasgupta, ''A multiagent swarming system for distributed automatic target recognition using unmanned aerial vehicles,'' *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 38, no. 3, pp. 549–563, May 2008.
[22] H. A. Kurdi, E. Aloboud, M. Alalwan, S. Alhassan, E. Alotaibi, G. Bautista, and J. P. How, ''Autonomous task allocation for multi-UAV systems based on the locust elastic behavior,'' *Appl. Soft Comput.*, vol. 71, pp. 110–126, Oct. 2018.
[23] H. A. Kurdi and J. P. How, ''Dynamic task allocation in an autonomous multi-UAV mission,'' U.S. Patent 344 014, Nov. 2019.
[24] A. Ahmad, S. Din, A. Paul, G. Jeon, M. Aloqaily, and M. Ahmad, ''Real-time route planning and data dissemination for urban scenarios using the Internet of Things,'' *IEEE Wireless Commun.*, vol. 26, no. 6, pp. 1536–1558. Dec. 2019.
[25] S. Zafar, S. Jangsher, O. Bouachir, M. Aloqaily, and J. Ben Othman, ''QoS enhancement with deep learning-based interference prediction in mobile IoT,'' *Comput. Commun.*, vol. 148, pp. 86–97, Dec. 2019.
[26] J. Rajawat, ''Genetic algorithms in search, optimization and machine learning (book review),'' *ORSA J. Comput.*, vol. 3, no. 2, p. 176, 1991.
[27] M. Mandelbaum, ''Flexibility in decision making: An exploration and unification,'' Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 1978.
[28] M. Li, M. Li, K. Yang, B. Xia, and C. Wan, ''A network-based portfolio optimization approach for military system of systems architecting,'' *IEEE Access*, vol. 6, pp. 53452–53472, 2018.
[29] B. Zhao, X. Wang, D. Lin, M. M. Calvin, J. C. Morgan, R. Qin, and C. Wang, ''Energy management of multiple microgrids based on a system of systems architecture,'' *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6410–6421, Nov. 2018.
[30] H. T. Tran, J. C. Domercant, and D. N. Mavris, ''Parametric design of resilient complex networked systems,'' *IEEE Syst. J.*, vol. 13, no. 2, pp. 1496–1504, Jun. 2019.
[31] D. M. Zelenović, ''Flexibility—A condition for effective production systems,'' *Int. J. Prod. Res.*, vol. 20, no. 3, pp. 319–337, May 1982.
[32] H. T. Tran, J. C. Domercant, and D. N. Mavris, ''Designing resilient system-of-systems networks,'' in *Proc. Annu. IEEE Int. Syst. Conf. (SysCon)*, Apr. 2017, pp. 1–6.
[33] S. Moulik, R. Devaraj, and A. Sarkar, ''COST: A cluster-oriented scheduling technique for heterogeneous multi-cores,'' in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2018, pp. 1951–1957.

TING DUAN received the bachelor's degree in information computing and science in 2018. She is currently pursuing the master's degree in management science and engineering with the National University of Defense Technology (NUDT), Changsha, China. Her research interests focus on system-of-systems architecture optimization, UAV cluster task scheduling, and systems simulation.

WEIPING WANG received the Ph.D. degree in systems engineering from the National University of Defense Technology (NUDT), Changsha, China, in 1998. He is currently a Professor with NUDT. His research interest focuses on systems engineering and simulation.

TAO WANG received the Ph.D. degree in software engineering from the National University of Defense Technology (NUDT), Changsha, China. He is currently an Associate Professor with NUDT. His research interests focus on systems engineering and simulation, multi-agent decision making under uncertainty, and data mining.



XIAOFAN CHEN received the Engineering degree in 2017. He is currently pursuing the master's degree in control science and engineering with the National University of Defense Technology, Changsha, China. His main research direction is deep learning.

XIAOBO LI (Member, IEEE) received the Ph.D. degree in control science and engineering from the National University of Defense Technology (NUDT), Changsha, China. He is currently an Associate Professor with NUDT. His research interests focus on systems engineering and simulation, multi-agent decision making under uncertainty, and SoS engineering.