Solving A Multi-objective Mission Planning Problem for UAV Swarms with An Improved NSGA-III Algorithm

Jiajie Liu [1], Weiping Wang [1], Xiaobo Li [1], Tao Wang [1,*], Senyang Bai [1] and Yanfeng WANG [1]

[1] College of Systems Engineering, National University of Defense Technology
137 Yanwachi, Changsha, Hunan 410073, China
E-mail: wangtao1976@nudt.edu.cn

Received 19 January 2017

Accepted 13 May 2018

Abstract

Restricted communication in unmanned aerial vehicle (UAV) swarms means that configuration needs to vary dynamically with changing tasks. We propose a mission planning model that uses a motif, a grouping of related functions, as the basic task unit. The planning model automatically generates a mission planning scheme from a task priority execution order given as an input. The selection of the best scheme from among possible solutions is a multi-objective optimization problem with calculation complexity rapidly increasing with the number of tasks. To address this difficulty, we enhance the NSGA-III algorithm by adding adaptive genetic operators when generating the offspring population. We apply the improved NSGA-III algorithm to optimize mission planning schemes with changing task priority execution orders. We validated the feasibility and effectiveness of the improved algorithm via a case study.

Keywords: mission planning; UAV swarms; motif; adaptive genetic operators; NSGA-III algorithm; optimization.

1. Introduction

Unmanned aerial vehicles (UAVs) have changed combat styles profoundly. Faced with an antagonistic, uncertain, and dynamic battlefield environment, the UAV combat style is evolving from a single platform operation to a networked swarming operation[1]. As such, mission planning is one key to UAV swarm operation.

Current mission planning methods for UAV swarms mainly focus on solutions to path planning problems[2–4] or reconnaissance mission planning problems[5–7]. These methods solve mission planning problems for UAV swarms effectively in some aspects but notably do not apply to combat mission planning problems, for two reasons. First, both path planning and reconnaissance mission planning apply to single function UAVs. UAV operation is a complex action that requires UAVs with different functions to coordinate with each other. Second, the existing mission planning methods do not take communication restrictions between UAVs into consideration. At the present time, technological restrictions greatly limit communication between UAVs. Strong electromagnetic interference from the battlefield enemy limits communication still further.

Further, the permutations and combinations of desired tasks produce a tremendous number of possible task priority execution orders. This number increases with the increase in the number of tasks. Selecting suitable planning schemes that meet requirements from this large pool of possible schemes has become a difficult problem. The last and most notable difficulty is that the choice of mission planning schemes is a multi-objective optimization problem. Having a large number of objectives slows down the search for a Pareto front, a result of increasing computational complexity. This optimization is on top of the already time-consuming complexity arising from each objective function itself. These time-consuming processes complicate real-time mission planning in a quickly changing confrontational environment.

To address these challenges, we propose a motif-based mission planning model. We group UAVs with different functions into motifs, each of which needs only a few communication connections. The motif is used as the basic task unit rather than an individual UAV. Our model generates a mission planning scheme automatically by using a multidimensional dynamic list scheduling (MDLS) algorithm that, in turn, uses a task priority execution order as input. Finally, we address the complexity problem by reducing the number of planning optimization indicators to three.

We introduce an evolutionary many-objective algorithm using a reference point-based non-dominated sorting approach (NSGA-III) to optimize planning schemes. NSGA-III is an advanced search technique offering better performance for multi-objective problems compared to other best-in-class MOEAs such as NSGA-II[8], MOEA/D[9] and MOEA/D-TCH[10]. To speed up iterative convergence, we apply adaptive genetic operators when generating offspring populations.

We also present our case study to demonstrate the feasibility and effectiveness of the improved NSGA-III algorithm.

The rest of this paper is organized as follows. Section 2 introduces related work. Section 3 describes the motif-based planning model and expresses it mathematically. Section 4 presents a customized and improved NSGA-III algorithm to solve the optimization problem. Experimental results are discussed in section 5. Section 6 gives our conclusions.

2. Related Work

Most of the existing research on UAV mission planning has focused on single function mission planning, such as path planning and reconnaissance mission planning. Koohifar et al.[11] introduced a receding horizon path planning algorithm for UAV swarms to localize a moving radio frequency transmitter cooperatively. Using this prediction model, they formulated the most favorable course of action to solve the path planning problem using local optimization, helping the whole system achieve the goal over a finite receding horizon. Choi et al.[12] proposed a method for utilizing an UAV swarm when the communication infrastructure is disabled due to war or natural disasters. The proposed method was validated and its performance evaluated using an NS-2 simulation.

Other researchers have begun to work on more complex mission planning. Slear[13] designed and implemented a comprehensive mission planning system that integrated several problem domains including path planning, vehicle routing, and swarm behavior. Lamont, Slear, and Melendez[14] designed and implemented a comprehensive mission planning system for swarms of autonomous aerial vehicles (UAV). Their system consisted of a parallel, multi-objective terrain path planner and a vehicle router. Both parts used evolutionary algorithms. Wei et al. [15] investigated the problem of complex dynamic mission planning for a UAV swarm and proposed a centralized-distributed hybrid control framework for mission assignment and scheduling. Boskovic et al.[16] investigated an autonomous hierarchical architecture for controlling swarms of UAVs to carry out complex missions. The approach effectively combined mission planning using evolutionary algorithms and biologically-inspired swarm behaviors. Through simulation and experiment, the system was shown to be resistant to intermittent communications problems and adaptable to dynamic changes in the environment. In [17], Sampedro C presented a scalable and flexible architecture for real-time mission planning and dynamic agent-to-task assignment for a UAV swarm. The proposal consisted of a Global Mission Planner (GMP) responsible for assigning and monitoring high-level missions through an Agent Mission Planner (AMP) in charge of handling each task at the UAV level.

Recent research applies swarm theory and methods in mission planning. However, existing research has not considered the communication restrictions of UAVs. In actual operation, communications cannot be maintained continuously due to complex working and electromagnetic environments. Thus, the system must change with communication ability and mission tasks.

3. Mission Planning Model

This section introduces our motif-based model that coordinates UAV operations by motif (related functions) rather than by individual UAV. Subsection 3.1 introduces 7 basic motifs to perform tasks and the mapping of tasks into motifs. Subsection 3.2 does mission decomposition. A complex mission is decomposed into detailed tasks. These tasks are performed by the 7 kinds of motifs that are proposed in subsection 3.1. In subsection 3.3, mission planning schemes are generated automatically by using MDLS algorithm based on subsection 3.1 and subsection

3.2. After presenting our model, subsection 3.4 expresses the model mathematically.

3.1 Motif-Based Configuration

3.1.1 Basic motifs to perform tasks

A motif is a subnet structure repeated in in one or more networks[18]. The motif idea is significant to our work as it serves as the functional unit in the framework [19]. Lee and Lee used motifs in the calculations for combat effectiveness[20].

We use motifs as basic task units in our planning model. Decision-making, information transmission, and operational functions are distributed among UAV swarm members. We divide UAVs into five categories based on function: DUAV (decision-making), RUAV (reconnaissance), AUAV (attack), RAUAV (reconnaissance and attack), and CUAV (communication relay).

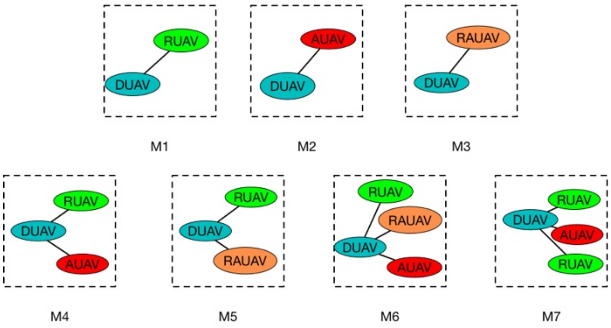

Fig. 1. Motif-based configuration diagram

Through observation and study of UAV operational networks, we have identified 7 types of significant motifs that appear with frequency. Fig. 1 depicts our groupings.

3.1.2 Tasks assigning to motifs

As communication plays an important role in UAV operation, we add a communication capability demand vector constraint. Fig. 2 shows the decomposition of task capability demand capability into operation capability demand and communication capability demand. We use the new capability demand vector to calculate the types and numbers of motifs needed to complete a task. A task is considered completable when the capability vectors of motifs is component-wise greater than or equal to the task's capability demand vector.

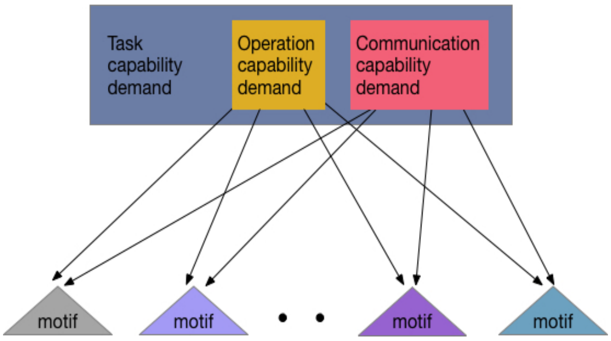

Fig. 2. Task requirement decomposition diagram

According to the characteristics of UAV operation, we propose a new operation capability demand vector which has six components: low altitude detection capacity, aerial reconnaissance capacity, sky-to-ground attack capacity, sky-to-sky attack capacity, patrol strike capacity and information processing capacity.

The six capabilities of the motif are defined based on the kind and number of UAVs, the capabilities of each UAV and the connection the motif involves. The ***j***th operation capability of motif ***i*** is β_{ij} .

$$\beta=\begin{bmatrix} \beta_{11} & \cdots & \beta_{17} \\ . & \cdots & . \\ \beta_{61} & \cdots & \beta_{67} \end{bmatrix} \tag{1}$$

n_k (the number of motif ***k***) must satisfy the following linear inequality:

$$\begin{pmatrix} n_1 & \ldots & n_7 \end{pmatrix} \bullet \beta^T \geq C_h \tag{2}$$

Communication demand vector has two components: the average degree of network and the clustering coefficient. The average degree of network ω is measured:

$$\omega = \frac{l}{n} \tag{3}$$

$$l = n_1 + n_2 + n_3 + 2n_4 + 2n_5 + 3n_6 + 3n_7 \tag{4}$$

$$n = 2n_1 + 2n_2 + 2n_3 + 3n_4 + 3n_5 + 4n_6 + 4n_7 \tag{5}$$

The average degree represents the average number of UAVs per UAV connected[21].

The clustering coefficient τ is measured by:

$$\tau = \frac{2l}{n(n-1)} \tag{6}$$

representing the probability of communication between the two UAVs[22].

Restrictions in bandwidth and communication distance limit the operation range of UAV swarms. CUAVs have wide bandwidth and long distance communication abilities. We use a CUAV to connect motifs performing the same task. Specifically, CUAVs connect to DUAVs in motifs. CUAVs are responsible for transmission of commands and the collection of battlefield information. Fig. 3 shows the communication structure in our model.

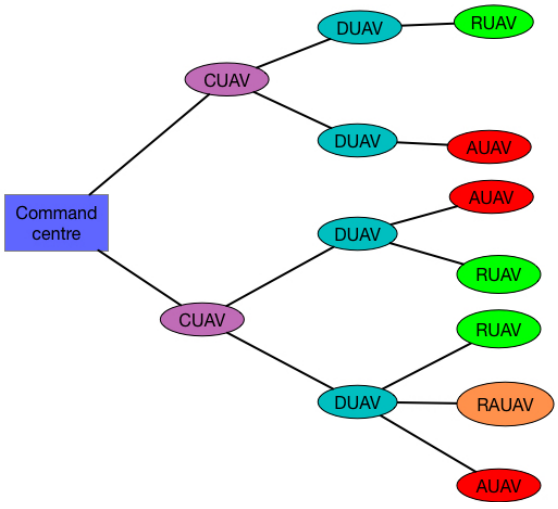

Fig. 3. Diagram of UAV communication connections

3.2 *Mission Decomposition*

The fundamental question underlying the design of a mission planning model is, "Who or what should do which part of the mission?" It implies that a mission is decomposable into a set of tasks. We use a goal decomposition method[23] to perform mission decomposition. We assume that a mission is decomposable into a task set $T_i(i=1,2,...,N_T)$, where N_T is the number of tasks. We characterized every task T_i according to the following attributes:

- estimated completion time $t_i(i=1,2,...N_T)$;
- motif demand vector $[m_{i1,}m_{i2,},...,m_{i7}]$, where m_{ij} is the number of the ***j***th motif required for successful completion of task T_i.

A mission is decomposable into a set of related tasks. Fig. 4 shows our use of directed task chains to detail the following correlations between tasks:

(1) Task priority[24]: In the case of insufficient resources, the task located at the front of the priority chain is to be carried out ahead of later tasks. In the case of sufficient resources, tasks can execute concurrently.

(2) Task precedence[25]: The placement of task T_i ahead of task T_j in a precedence chain means T_j cannot start until T_i completes.

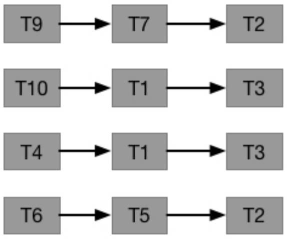

Fig. 4. Task priority chains graph

We treat task precedence correlations as special priority relations with strict restrictions in time. A mission planning scheme is feasible only if its task priority execution order fits all task priority chains.

3.3 *Mission Planning*

By decomposing the mission and assigning tasks to motifs, we obtain the types and numbers of motifs required for each task. According to the existing UAV resources and the needed, we use our MDLS algorithm to generate mission planning schemes automatically.

3.3.1 Planning scheme generation with MDLS

MDLS is often used in combat to schedule equipment and generate mission planning schemes, achieving good results[26 27]. MDLS contains two main steps.

Step 1: Select the next ready task for processing. A task becomes ready when all tasks that have a priority over it have started and all its predecessors have been completed.

Step 2: Select usable UAVs to be assigned to the motifs required for the task. A usable UAV is one that is not performing a task.

The MDLS algorithm generates a mission planning scheme automatically upon receiving task execution priorities. The algorithm selects tasks one by one according to the task priority execution chain and then generates a mission plan automatically. Algorithm 1 presents the algorithm in full.

3.3.2 Scheme evaluation index

Our strategy uses three operational effectiveness indexes to evaluate mission planning schemes.

(1) Mission completion time (MCT): This is a common index of operational effectiveness.

(2) The average number of altered connections (ANAC): Topology changes within a UAV network contribute

to communication delays[28]. A decrease in the number of altered connections reduces communication delays and enhances the stability of the UAV network.

(3) The average number of working UAVs (ANWU): During operations, UAVs are easily destroyed. When UAVs are damaged and unable to continue to operational tasks, spare UAVs should replace the damaged UAVs.

Algorithm 1: MDLS algorithm

```
Input:  the set of tasks that have been completed $T_C = \varnothing$; the set of tasks that have started, $T_S = \varnothing$;
        the set of ongoing tasks' completion times $FT = \{0\}$;
Output: mission planning scheme

1  $f = \min_{ft \in FT}(ft)$ ($FT$ is a set of ongoing tasks' completion times)
2  $FT \leftarrow FT \setminus \{f\}$
3  $t_c \leftarrow t_c \cup \{f\}$ ($t_c$ is a collection of all task completion times)
4  $Spare \leftarrow Spare + m(F_G)$ ($F_G$ are the corresponding groups of tasks, $Spare$ is a set of spare UAVs)
5  $T_C \leftarrow T_C \cup F_G$
6  $READY$ Update ($READY$ is a set of ready tasks.)
7  if $\forall i \in READY$ cannot satisfy $D \bullet m(T_i)' \leq Spare$ ($m(T_i)$ is the motif demand vector of $T_i$)
8      Go to Line 1.
9  else Go to Line 11
10 end if
11 if $READY = \varnothing$
12     Go to Line 1
13 end if
14 $t_c \leftarrow t_c \cup \{f\}$
15 $READY1 = \{i \in READY | [n(F_T), D \bullet m(T_i)'] \leq Spare\}$
16 Select $i \in READY1$
17 $READY \rightarrow READY \setminus \{i\}$
18 $T_S \leftarrow T_S \cup i$
19 $FT \leftarrow FT \cup (f + t_i)$ ($t_i$ is the processing time of $T_i$)
20 Select $m(T_i)$ from $Spare$
21 $Spare \leftarrow Spare\text{-}m(T_i)$
22 do until $D \bullet m(T_i)' \geq Spare$, $\forall i \in READY$
23     Go to Line 11
24 end do
25 if $M \bullet m(T_i)' \geq Spare, \forall i \in READY$
26     Go to Line 1
```

3.3.3 Dynamic Reconfiguration

The MDLS algorithm requires three conditions to be met to begin a task. First, all tasks with higher priority than the target task must have started. Second, all of the target task's predecessors must have completed. Third, existing UAV resources must meet the demands of the task.

The MDLS algorithm selects tasks one by one according to task priority execution order. As long as the priority execution order fits all task priority chains, i.e. the mission planning scheme is feasible, the first condition is satisfied. When the second and third conditions are satisfied, the task can start. UAV resources are released only when ongoing tasks are completed. When considering the first or second conditions, the starting time of a task is 0 or the time of completion of an earlier task. Changing tasks requires reorganization of communication connections to suit the upcoming task.

We use existing motifs as much as possible to reduce communication delays. During a changeover, all CUAVs disconnect from previous motifs and connect to new motifs. As a result, the number of altered connections attached to the CUAVs is constant and not counted in the number of altered connections. Fig.5 shows the persistence of previous motifs required for upcoming tasks. Motifs that are unneeded in the upcoming task disconnect their internal connections to allow for the new.

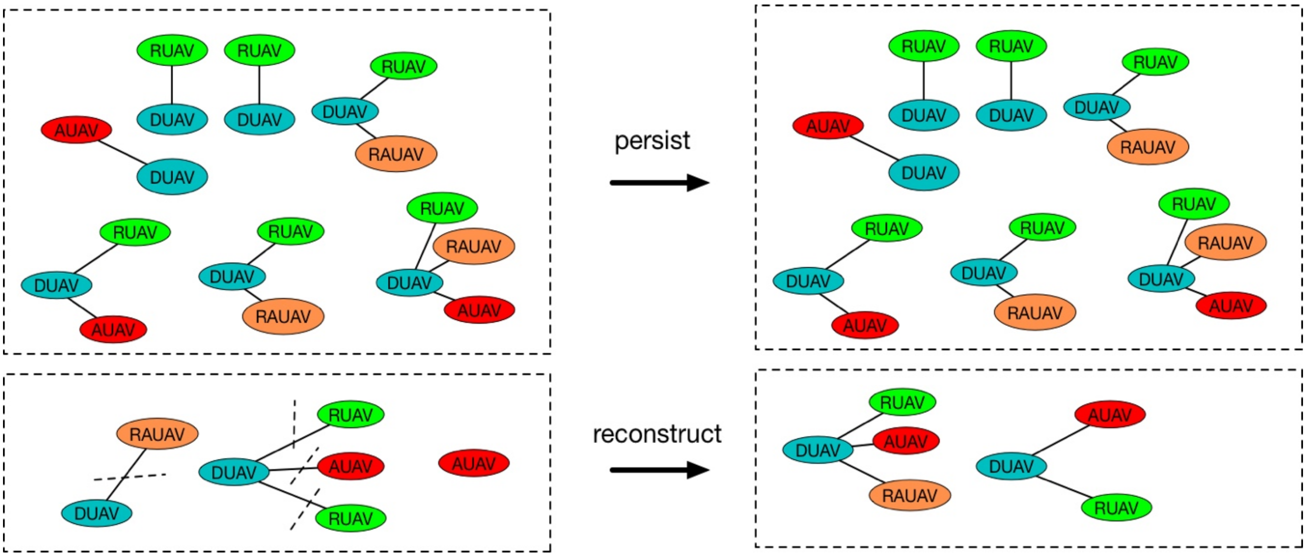

Fig. 5. Preservation and reconstruction of motifs graph.

3.4 *Mathematical Model of Mission Planning*

3.4.1 Optimization variable

Once the priority execution order of tasks is determined and used as an input into MDLS algorithm, our algorithm automatically generates a mission planning scheme after calculating the three scheme evaluation indexes. Our mathematical model also includes the priority execution order of tasks as variables, as a planning scheme is feasible only if its task the priority execution order fits all priority chains.

3.4.2 Objective functions

As is mentioned in Section 3.3.1, there are three objective functions:

1. MCT: It is equal to the finishing time of the last completed task: $t = \max(FT)$.
2. ANAC: Communication connections change when a task completes. If no task is immediately ready, the motif disconnects all its connections to prevent enemy snooping.

   We calculate the number of altered connections for a single task according to

$$n_r(t_k) = \left| m_c(t_k) - m_s(t_k) \right| \bullet (1,1,1,2,2,3,3)^T \quad (7)$$

   where $m_c(t_k)$ is the total motif vector of tasks completed at time t_k, $m_s(t_k)$ is the total motif vector of tasks starting at time t_k (if no task starts, $m_s(t_k)$ is 0). The number of total altered connections is

$$n = \sum_{t_j \in t_c} n_r(t_k) \quad (8)$$

   where t_k is a collection of all task completion times.

3. ANWU: The number of working UAVs changes when a task completes. $T = t_s \cup t_e$ is the set of all times, ordered from smallest to largest in the set. The number of working UAVs from $T(i)$ to $T(i + 1)$ is

$$s(i) = \sum_{\substack{t_s(k) \le T(i) \\ t_e(k) \ge T(i+1)}} m_k \bullet (2,2,2,3,3,4,4)^T \quad (9)$$

The average number of working UAVs is

$$s = \frac{1}{n(T)} \sum_{i=1}^{n(T)} s(i) \qquad (10)$$

4. Planning Scheme Optimization through an Improved NSGA-III Algorithm

The MDLS algorithm generates a mission planning scheme from the task priority execution order. The computational complexity increases dramatically with the number of tasks and their related priorities.

To efficiently solve the mission planning problem, we use a multi-objective evolutionary algorithm (MOEA) heuristic called NSGA-III, which has been demonstrated to outperform most MOEA approaches on multiple objective benchmarks.[29]. We also apply adaptive genetic operators in the generation of offspring population and propose an improved NSGA-III algorithm for better performance in this application. Specifically, the probability of crossover operation and mutation are no longer constant. Instead, we adjust them according to the non-dominated sorting levels of chromosomes. Chromosomes with high rankings in the sorted population have a higher probability of crossover and lower probability of mutation, which helps preserve good chromosomes. The use of adaptive genetic operators accelerates the determination of the Pareto set.

The following subsections present the components of our algorithm. Additional details are available in [29].

4.1 Framework of the Algorithm

NSGA-III algorithm is an elitist (μ+λ) approach, as is shown in Fig. 6. It can preserve the elite population members.

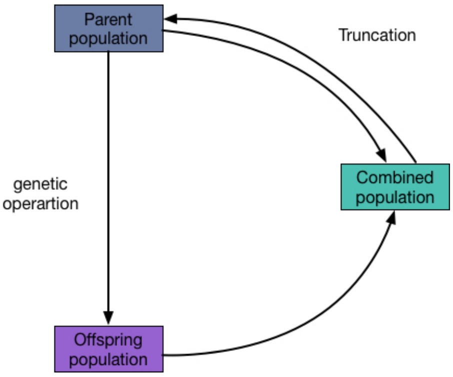

Fig. 6（μ+λ）elitist framework.

The NSGA-III algorithm applies the general framework of the NSGA-II algorithm[30], which consists of population initialization, reference point generation, genetic operators, non-dominated sorting, adaptive normalization, association operations, and niche-preservation operations, as is seen in Fig.7.

We use two adaptive functions to indicate the probability of crossover and mutation. Population members which perform well have a higher probability of crossover and lower probability of mutation, which helps preserve good chromosomes. Because the probability of the crossover operation and mutation of a chromosome is decided by its nondominated sorting level, we add non-dominated sorting of population ***P*** before the crossover operation and mutation.

Unlike other MOEAs, population sorting and selection among population members in the NSGA-III algorithm is aided by supplying and adaptively updating a number of well-spread reference points on a normalized hyperplane.

Initially, random ***N*** candidate population members, ***P*** and ***H*** reference points, ***R*** are generated. The candidate population members and reference points are then evolved for a fixed number of generations. In each generation *t*, parent population P_t is first classified into non-dominated levels. Then P_t is subjected to adaptive genetic operators to produce ***N*** offspring C_t according to parent chromosomes' non-dominated levels. P_t and C_t are then pooled respectively, and the combined population is classified into non-dominated levels. We select population members from level 1 until the number of selected population members is greater than or equal to ***N***. If the number of population members from level 1 to level ***l*** exactly equals ***N***, no further operations are needed. If the number is greater than *N* from level ***l***, then we sort population members and choose ***k*** members of them from level ***l*** according to their association with the reference points.

It is worth noting that each solution is a task priority execution order which is subjected to all task priority chains. First, we must guarantee that each solution in the initial population is feasible, i.e., conforming to all task priority chains. As a result, we design a solution generator to generate the eligible initial population. During the genetic process, in order to prevent producing ineligible solutions in the offspring, we use a segment crossover operator and introduce a heuristic mutation operator.

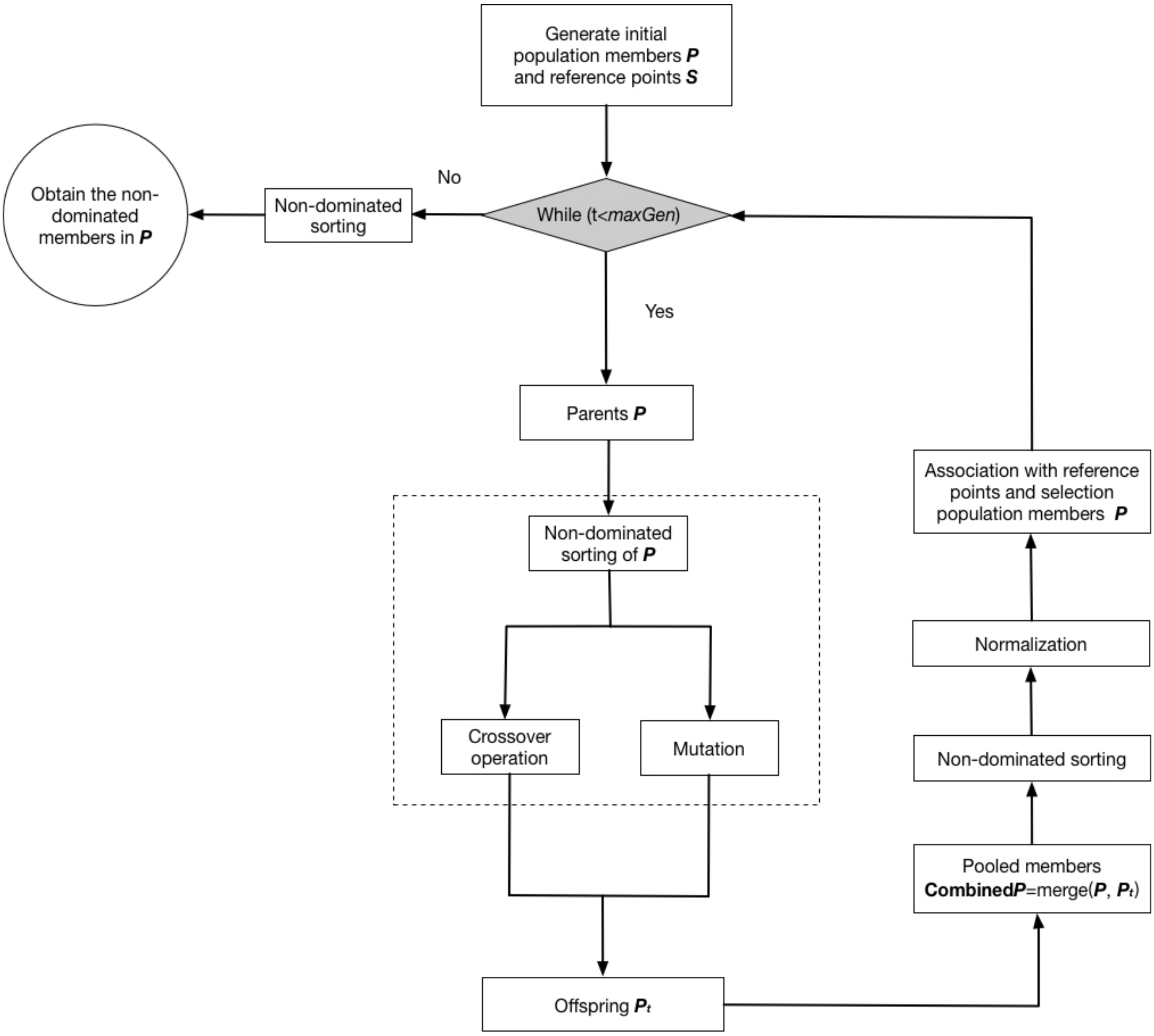
Generate initial population members *P* and reference points *S*
No
Obtain the non-dominated members in *P*
Non-dominated sorting
While (t<*maxGen*)
Yes
Parents *P*
Association with reference points and selection population members *P*
Non-dominated sorting of *P*
Normalization
Crossover operation
Mutation
Non-dominated sorting
Pooled members **CombinedP**=merge(*P*, *Pt*)
Offspring *Pt*

Fig. 7. The improved NSGA-III framework graph

We present the improved NSGA-III algorithm in Algorithm 2.

Line 1 initializes the offline archive ***BestF*** to the empty set ∅.

Line 2 initializes level ***l*** population members needing association with reference points and selection to 1.

Line 3 initializes population members on from level 1 to level ***l***-1 to empty set ∅.

Line 4 initializes population members on level ***l*** to empty set ∅.

Lines 5 and 6 generate the initial population ***P*** conforming to all task priority chains and the initial reference point set ***R***.

Line 8 calculate objective functions of the current population.

Line 9 performs a non-dominated sorting operation on the population ***P***, dividing population members into different non-dominated levels.

Lines 10 and 11 perform the crossover operation on the population and calculate the objective functions for the resulting offspring population.

Lines 12 and 13 perform the mutation operation on the population and calculate the objective functions for the offspring population resulting from mutation.

Lines 14 through 16 sorts the combined population members on different levels, adaptively normalizes the population members from levels 1 to ***l***, and selects ***N*** population members from members on levels 1 through level ***l*** through their association with reference points.

Line 18 performs a non-dominated sorting operation on the population of the last generation ***P*** and archives non-dominated population members in ***BestF***.

Algorithm 2. The improved NSGA-III algorithm

```
Input: Initial population members, P of size N, maximum number of generations,
       maxGen, the number of objectives, M, the number of divisions along each
       objective, p, reference points, R of size H, crossover probability, pc, mutation
       probability, pm
Output: P, BestF
1 BestF ← ∅;
2 l=1;
3 P1 ← ∅; (P1 are population members on level 1 to level l)
4 P2 ← ∅; (P2 are population members on level l)
5 P ← generatepopulation(N);
6 R ← generatereferencepoints(M,p);
7 for t ← 1 to maxGen
8     F_P ← objectiveFunction(P);
9     Non-dominated sorting(P);
10    Oc ← Crossover(P,pc);
11    Fc ← objectiveFunction(Oc);
12    Om ← Mutate(P,pm);
13    Fm ← objectiveFunction(Om);
14    (JointP,JointF) ← multisetUnion(P,Oc,Om,F,F_c,F_m);
15    (P1,l,P2,K) ← nondominatedsorting(JointP,JointF);
16    P2N ← normalization(P1,P2);
17    P ← Associationandselection(P2N,K,R);
18 end
19 BestF ← nondominatedsorting(P);
```

4.2 Eligible Population Generator

Members in a population must conform to the orders in all task priority chains. First, the input population must be feasible, i.e., with all members meeting all task priority chains. To ensure this, we randomly select a ready task over that tasks have a priority over have been selected out. Next, we update all task priority chains by removing the selected task. We then select a ready task and update the task priority chains again until all tasks are selected out. Finally, we generate an eligible task priority execution order. This process is repeated until an initial population conforming to all task priority chains is filled.

4.3 Crossover Operator

This operator performs crossover determinations on the chosen parent chromosomes to generate the offspring chromosomes. First, we select a random pair of chromosomes from the population. We use a varying probability pc determined by.

$$pc = \frac{1}{2^{l_1-1} \cdot 2^{l_2-1}} \qquad (11)$$

where ***l1***and ***l2*** are the non-dominated levels of the two chromosomes. They are calculated during the non-dominated sorting of population ***P***. By including the adaptive probability, chromosomes which perform well are more likely to crossover. This speeds up convergence and determination of the Pareto set.

In order to keep chromosomes feasible after crossover operation, we use a segment crossover operator. The middle point located at $\lceil N_T / 2 \rceil$ divides parent chromosomes into two gene segments. Gene segment 1 is from 1 to $\lceil N_T / 2 \rceil - 1$, and gene segment 2 is from $\lceil N_T / 2 \rceil$ to N_T. The new offspring chromosome is constructed by leaving gene segment 1 unchanged and constructing gene segment 2 based on the sorting of these tasks in another parent chromosome. This is known as a segment crossover, which maintains the relative orders of tasks in chromosomes. Fig. 8 provides an example of the crossover operator. Chromosome 1 is on non-dominated level 2 and chromosome 1 is on non-dominated level 3. The probability of crossover operation is $pc = \frac{1}{2^{2-1} \cdot 2^{3-1}} = \frac{1}{8}$.

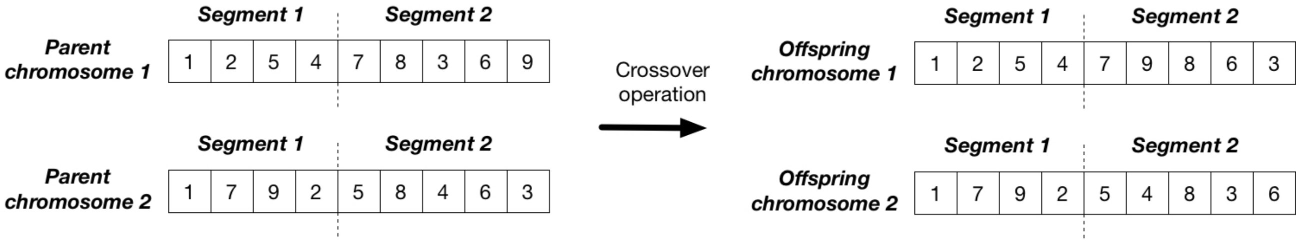

Fig. 8. Crossover operation example

4.4 *Mutation Operator*

This operator performs mutation on the chosen parent chromosome to generate an offspring chromosome. First, a percentage ***mu*** of chromosomes are selected out one by one from Pc for mutation. When a chromosome is selected, a random number is generated. If this number is less than or equal to ***mu***, a mutation operation is performed on this chromosome. Otherwise, processing advances to the next chromosome.

After a chromosome is selected out, the probability of the chromosome mutation is determined by.

$$pm = \frac{1}{2^{l-1}} \qquad (12)$$

where ***l*** is the non-dominated level of the chromosome. Population members whose non-dominated ranking is near the front have a smaller probability of mutation. The adaptive mutation operator attempts to preserve solutions that perform well.

This mutation operator performs mutation by exchanging points. The algorithm determines whether or not there is a point that can be exchanged with a selected point. If neither the point before nor the point after can be exchanged with the selected point, mutation of the selected point cannot occur. In that event, we randomly select another point and repeat the test until reaching a point which can be exchanged. This point is indicated by ***pi***.

After selecting ***pi***, we calculate the exchangeable range of ***pi***. We select all previous tasks of ***pi*** in all priority chains. If it did not exist, it is denoted as 0. We archive these points in the previous task set ***Bi***. Similarly, we select the next task for ***pi*** and, if it did not exist, denote it as $N_T + 1$. We archive these points in the next task set ***Ai***. The range [***b***,***a***] (***b*** $\in$ ***B***, ***a*** $\in$ ***A***), represents a gene segment from ***b*** to ***a*** in the chromosome. The exchangeable range of ***pi*** is [***bi***,***ai***], which contains the fewest points in the chromosome. To ensure a diversity of mutation, we first determine all points in [***bi***,***ai***] that could be exchanged with ***pi*** and then select one of them randomly. Algorithm 3 shows the pseudocode steps.

Figure 9 provides an example of the mutation operator. The task priority chains are 2-1-6-7, 2-4-3, and 8-6-5-9. We randomly choose 4 and find the set {1,8,6} which can be exchanged with 4. Next, we choose 8 for the exchange and obtain a new offspring chromosome. The chromosome's non-dominated level is 3. The probability of mutation is $pm = \frac{1}{2^{3-1}} = \frac{1}{4}$.

Algorithm 3. Function mutation operator (***P, pc, mu***)

```
    Input: Initial population members, P of size N, mutation percentage, mu, mutation
           occurring probability, pm, priority order chains, Tor
    Output: Offspring Om
 1 Pc ← popselection(P, pm);
 2 t=1;
 3 while t N pc
 4   k ← random(0,1);
 5   if k>mu
 6     Om(t) ← chromosome(t);
 7     k=k+1;
 8   else pi ← rand(random($(0,N_T)$));
 9     if pi is not exchangeable
10       Go to line 7;
11       else C ← exchangeableset(chromosome(t),Tor,pi);
12       q ← rand(random($(0,|C|)$));
13       Om(t) ← exchange(chromosome(t),pi,C(q));
14     end
15   end
16 end
```

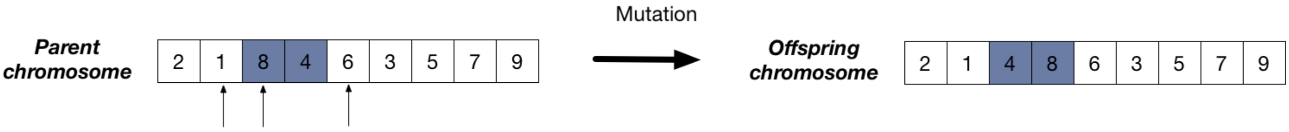

Fig. 9. Mutation operator example

5. Case Study

We performed our experiment with a mission consisting of 6 CUAVs, 20 DUAVs, 19 RUAVs, 17 AUAVs, and 12 RAUAVs. The percentage of each type of UAVs has important impacts on the results. It will greatly limit the concurrent completion of tasks. It is a complicated process to study the relation between the percentage of UAVs and the results. The case study is based on the fixed number of UAVs.

The mission included capturing a power station, capturing a waterworks and destroying a command center. The mission geographic layout is shown in Fig.10. We used the goal decomposition method to decompose the mission into 15 detailed tasks, as is shown in Fig.11.

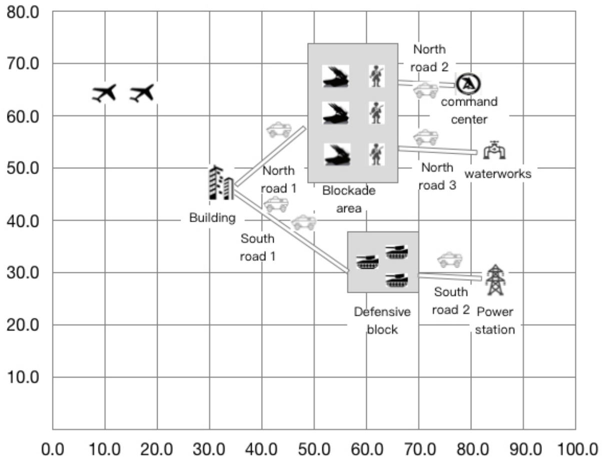

Fig. 10. Geographic constrains for the case.

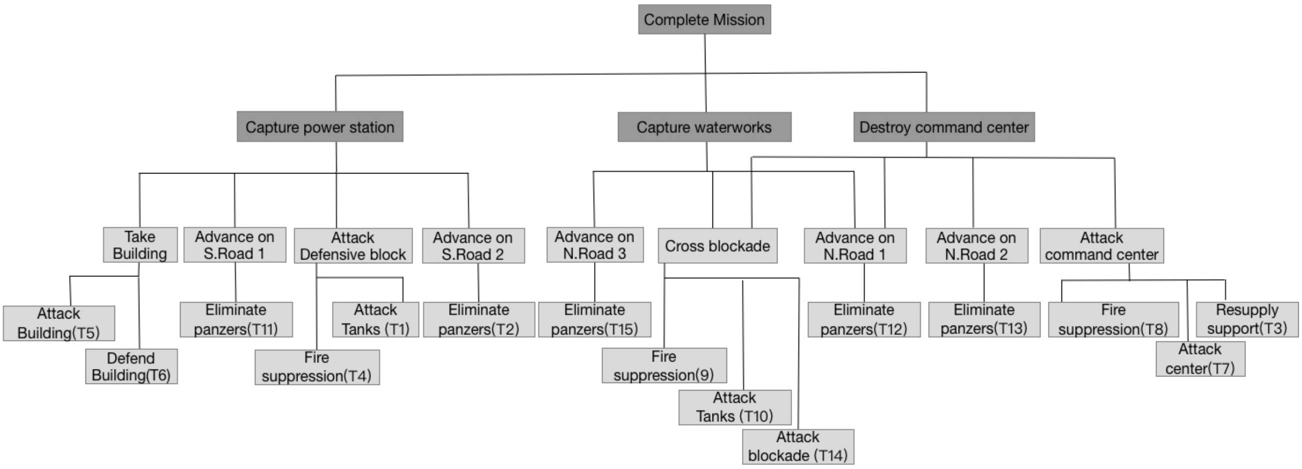

Fig. 11. Mission decomposition diagram.

These tasks have two kinds of correlations among them: task priority and precedence. We calculated the numbers and types of motifs required to complete each task as given in Table 1 along with the task estimated times. The task priority chains were T9-T10-T14-T15, T4-T1-T2, and T8-T7-T3. The task precedence orders were T7-T3 and T14-T10.

Table 1. Detailed information of tasks.

	motif	M1	M2	M3	M4	M5	M6	M7	time (min)
task	T1	1	0	3	2	0	1	2	15
	T2	3	1	1	2	1	0	0	8
	T3	3	3	0	2	3	1	1	7
	T4	2	3	3	1	2	1	1	12
	T5	1	2	1	2	0	1	1	11
	T6	1	1	2	1	0	1	0	6
	T7	2	0	1	2	1	0	1	8
	T8	3	3	2	3	4	3	2	6
	T9	4	2	3	3	2	1	2	7
	T10	2	1	2	1	0	2	1	10
	T11	1	2	1	3	1	2	1	11
	T12	2	3	1	1	1	0	1	8
	T13	3	0	1	2	2	1	0	10
	T14	1	1	2	0	1	0	0	12
	T15	2	0	0	1	1	1	1	15

As we explained in the introduction, the selection of the best mission strategy is a difficult problem. We wanted to optimize the three combat effectiveness indexes simultaneously. We encoded the solutions and used the improved NSGA-III algorithm to solve this problem using the parameters given in Table 2.

We applied both the improved and original algorithms to solve this problem. Figs. 10 provides the 3D and 2D results of the last generation got by two algorithms. Red circles represent the last generation obtained from the original NSGA-III algorithm.

Table 2. Algorithm parameters

Parameters	**Values**
Population size ***N***	100
Maximum generations ***maxGen***	50
Divisions on each axis	10
Mutation percentage ***mu***	0.3

Blue stars (*) represent the last generation from the improved NSGA-III algorithm. Figure 12 includes a 3D representation of the population members of the last generation, showing three objects. As shown in the 3D figure in Figure 10, blue * are nearly covered by red circles, i.e. blue stars are dominant over red circles. Population members in the last generation obtained by the improved NSGA-III are dominated over those obtained by the original NSGA-III algorithm. Figure 12 also depicts a 2D figure of the population members where each represents 2 objectives of the 3. These figures show that solutions can achieve better combat effectiveness obtained by the improved NSGA-III algorithm than the original NSGA-III algorithm. In summary, the improved NSGA-III algorithm is more efficient than original NSGA-III algorithm.

From Fig.12, we also observe that the mission completion time had a significant negative relationship with the average number of used UAVs. In other words, a mission planning strategy for UAV swarms which causes fewer communication delays will cost more time. Based on the specific requirements of a battle, commanders can choose a mission plan from the non-dominated points. For example, if commanders give priority to completion time, it should be the first parameter taken into consideration to ensure the completion time of the mission, with the other functions optimized as much as possible beyond that.

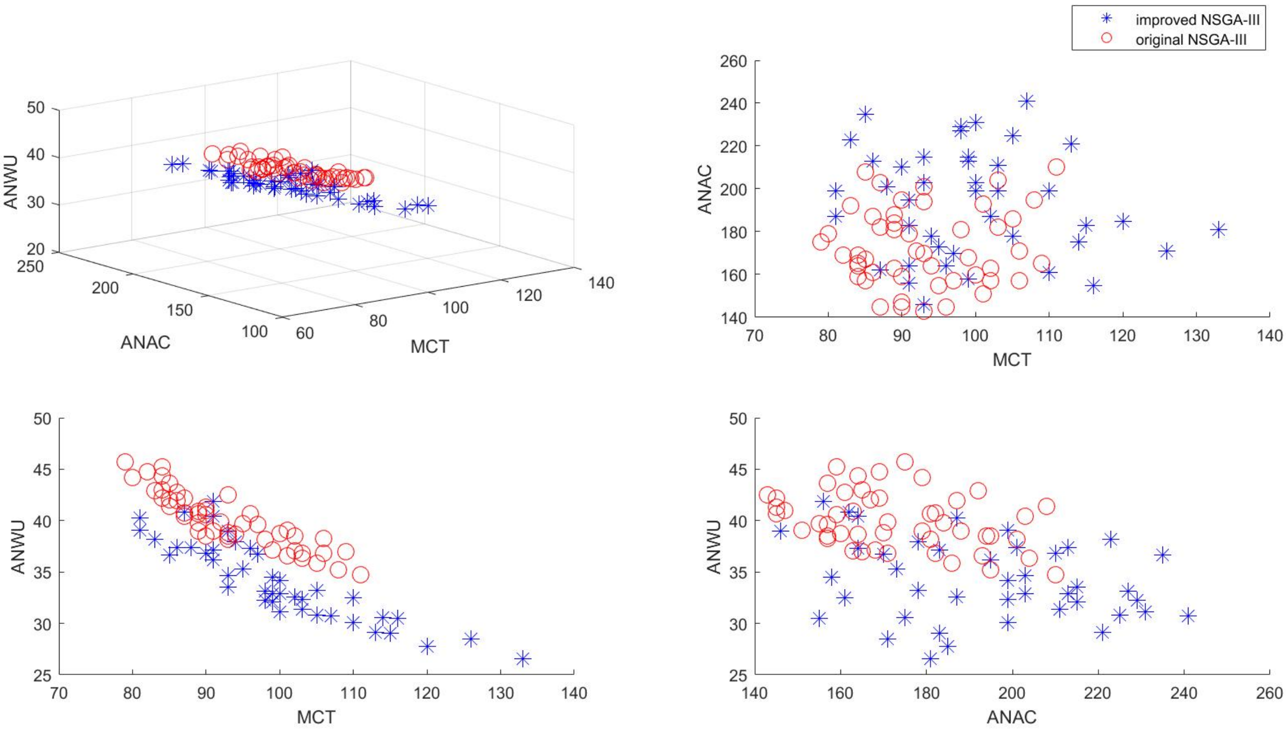

Fig. 12. Comparisons of Pareto front of the last generation got by two algorithms

We used the simple case study to validate the feasibility and effectiveness of the proposed mission planning method. In the case study, the number of UAVs is small and the percentage of each type of UAVs is fixed. The method is also applicable when the number of UAVs grows or the percentage of UAVs changes. In the paper, we did not take war loss of UAVs into consideration. When considering the war damage, we should use a time dependent function related to the war loss instead of a fixed number to indicate the total number of UAVs. We will continue to study it in our future work.

6. Conclusions and Future Work

In this paper, in order to solve the mission planning problem under conditions of limited communication, we proposed a mission planning model using motifs as the basic task units. Furthermore, given the task execution ordering, we used the MDLS algorithm to schedule UAVs. As a result, selecting a good task execution order was key to a mission planning strategy. To achieve mission goals optimally, we used an improved NSGA-III algorithm optimize task execution order. Our process obtains a set of non-dominated solutions from which commanders can choose the most adequate one. We demonstrated the feasibility and effectiveness of the algorithm through a case study validation.

Due to time limitations, there are some shortcomings to our study. First, we simplified calculations by disconnecting all CUAVs from motifs at task completion. Furthermore, during a connection change, redundant motifs disconnect all internal connections, and single UAVs construct new motifs. These two practices do minimize the number of altered connections. Second, the percentage of each type of UAVs in the case study is fixed. Third, we did not take war loss of UAVs during combat process into consideration.

In the future, we plan to use an optimization algorithm to minimize the number of altered connections during the reconfiguration process. Furthermore, we would like to study the relation between the percentage of UAVs and the scheme evaluation indexes through setting contrastive experiments. In addition, we would take war loss of UAVs into consideration. We will find appropriate functions to represent the war loss of UAVs. We will replace fixed numbers with UAV number functions to indicate numbers of each type of UAVs in MDLS algorithm.

Conflict of interest

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

References

1. Orfanus, D.; Freitas, E. P. D.; Eliassen, F. Self-organization as a supporting paradigm for military UAV relay networks. *IEEE Communications Letters* **2016**, *20*, 804-807.
2. Liu W, Zheng Z, Cai K Y. Bi-level programming based real-time path planning for unmanned aerial vehicles[J]. *Knowledge-Based Systems* **2013**, 44(44):34-47.
3. Zheng Z, Wu S, Liu W, et al. A feedback based CRI approach to fuzzy reasoning[J]. *Applied Soft Computing* **2011**, 11(1):1241-1255.
4. Zheng, Z.; Liu, Y.; Zhang, X. The more obstacle information sharing, the more effective real-time path planning? *Knowledge-Based Systems* **2016**, *114*, 36-46.
5. Manyam, S. G.; Rasmussen, S.; Casbeer, D. W.; Kalyanam, K.; Manickam, S. Multi-UAV routing for persistent intelligence surveillance & reconnaissance missions. International Conference on Unmanned Aircraft Systems, Miami, USA, 27 Feb 2017.
6. Yang, W. L.; Lei, L.; Deng, J. S. Optimization and improvement for multi-UAV cooperative reconnaissance mission planning problem. International Computer Conference on Wavelet Active Media Technology and Information Processing, Chengdu, China, 19-21 Dec 2014, pp.10-15.
7. Shi, Z.; Huang, X.; Hua, Y.; Xu, D. Statistical physics method for multi-base multi-UAV cooperative reconnaissance mission planning. Advanced Information Technology, Electronic and Automation Control Conference. Chongqing, China, 19-20 Dec 2015 pp.64-68.
8. K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, “A fast and elitist multi-objective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, **2002**, 6(2) : 182–197.
9. Zhang Q, Li H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition[J]. *IEEE Transactions on Evolutionary Computation*, **2007**, 11(6):712-731.
10. Deb, K.; Jain, H. An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, part i: solving problems with box constraints. IEEE Transactions on Evolutionary Computation 2013, 18, 577-601.
11. Koohifar, F.; Kumbhar, A.; Guvenc, I. Receding Horizon Multi-UAV Cooperative Tracking of Moving RF Source, *IEEE Communications Letters* **2016**, 21, 1433-1436.
12. Choi, H. H.; Su, H. N.; Shon, T. Two tier search scheme using micro UAV swarm. *Wireless Personal Communications* **2017**, 93, 349-363.
13. Slear J N. AFIT UAV swarm mission planning and simulation system /[J]. **2006**.
14. Lamont, Gary B., J. N. Slear, and K. Melendez. "UAV Swarm Mission Planning and Routing using Multi-Objective Evolutionary Algorithms." *Computational Intelligence in Multicriteria Decision Making, IEEE Symposium on* IEEE, **2007**:10-20.

15. Wei, Y.; Blake, M. B.; Madey, G. R. An operation-time simulation framework for UAV swarm configuration and mission planning. *Procedia Computer Science* **2013**, 18, 1949-1958.
16. Boskovic, J.; Knoebel, N.; Moshtagh, N.; Amin, J.; Larson, G. Collaborative Mission Planning & Autonomous Control Technology (CoMPACT) System Employing Swarms of UAVs. AIAA Guidance, Navigation, and Control Conference, 10 - 13 August 2009, Chicago, Illinois.
17. Sampedro C, Bavle H, Sanchez-Lopez J L, et al. A flexible and dynamic mission planning architecture for UAV swarm coordination[C]// International Conference on Unmanned Aircraft Systems. IEEE, 2016:355-363.
18. Milo, R.; Shenorr, S.; Itzkovitz, S.; Kashtan, N.; Chklovskii, D.; Alon, U. Network motifs: simple building blocks of complex networks. Science **2002**, 298.
19. Pan, G.; Tang, J.; Guo, F. Analysis of co-associated transcription factors via ordered adjacency differences on motif distribution. *Scientific Reports* **2017**, 7, 43597.
20. Lee, Y.; Lee, T. Network-based metric for measuring combat effectiveness. *Defence Science Journal* **2014**, 64, 115-122.
21. Tang, C. L.; Wang, W. X.; Wu, X. & Wang, B. H. Effects of average degree on cooperation in networked evolutionary game. EPJ B. J. **2006**, 53(3), 411-415.
22. Gerhardt, G. J. L.; Lemke, N. & Corso, G. Network clustering coefficient approach to DNA sequence analysis. CHAOS SOLITON FRACT. J. **2006**, 28(4), 1037-1045.
23. Müller, H.; Sternad, D. Decomposition of variability in the execution of goal-oriented tasks: three components of skill improvement. *Journal of Experimental Psychology Human Perception & Performance* **2004**, 30, 212-233.
24. Bate, I. J. Limited pre-emptive global fixed task priority. IEEE 34TH REAL-TIME SYSTEMS SYMPOSIUM (RTSS 2013), 7975, 182-191.
25. Shehory, O.; Kraus, S. Formation of Overlapping Coalitions for Precedence-Ordered Task-Execution Among Autonomous Agents C. Int Conf on Multiagent Systems, 1996.
26. Yang, D. S.; Lu, Y. L.; Liu, Z.; Zhang, W. M. Research on algorithms of task scheduling. International Conference on Machine Learning and Cybernetics, Shanghai, China, 26-29 Aug 2004, pp.42-47.
27. Wang, J. F.; Bao, W. D.; Zhang, M. J. A Novel Task Resource Distribution Algorithm Based on MDLS and GA. International Conference on Machine Learning and Cybernetics, Hong Kong, China, 19-22 Aug 2007, pp.890-895.
28. Fu, X. M.; Zhao, M. Y.; Li, Y.; Yao, X. M. The Formation Method of Dynamic Alliance of UAV Communication Network. *Systems Engineering and Electronics* **2017**, 39, 193-197.
29. Deb, K.; Jain, H. An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, part i: solving problems with box constraints. *IEEE Transactions on Evolutionary Computation* **2013**, 18, 577-601.
30. Deb, K. A fast elitist multi-objective genetic algorithm: nsga-ii. IEEE Transactions on Evolutionary Computation 2000, 6, 182-197.