Deadlock-free and path coordination algorithm for Cooperative Multi-task Allocation Problem with Target Priority Constraints

Yiyang Zhao¹, a, Deyun Zhou², b, Xiaoyang Li¹, c, Zhen Yang², d, Kai Zhang², e, Lina Zeng², f

¹School of Electronics and Information, Northwestern Polytechnical University, Xi’an, China
a zhaoyiyang@mail.nwpu.edu.cn
b dyzhou@nwpu.edu.cn
c lixiaoyang@nwpu.edu.cn
d nwpuyz@mail.nwpu.edu.cn
e zhangkainwpu@mail.nwpu.edu.cn
f zenglina@nwpu.edu.cn

Abstract—In this paper, we propose a cooperative multi-task allocation problem with target priority constraints (CMTAPTPC) for unmanned aerial vehicles (UAVs). For solving the deadlock problem, a deadlock detection and elimination method, including target priority constraints, is introduced in the graph representation framework. Furthermore, an innovative waitable path coordination (WPC) algorithm is proposed to coordinate the flight path that violates time constraints. It adopted some strategies to reduce the complexity of coordination and save the energy of the UAVs. In the simulation part, a representative complex scenario is designed to illustrate the rationality and effectiveness of the CMTAPTPC model and WPC algorithm.

1. INTRODUCTION

Unmanned aerial vehicle (UAV) technology has developed enormously in recent years, and it is widely used in modern information warfare to perform surveillance, reconnaissance, attack, battlefield assessment, and decoy tasks [1]. However, the low survivability and limited payload of individual UAVs make it challenging to adapt to complex scenarios (e.g., wide-area search for munitions [2] and suppression of enemy air defenses [3]). Therefore, it is inevitable to use heterogeneous UAVs to cooperate in complex tasks in the future, i.e., UAVs with different capabilities, payloads, and maneuverability.

Due to the complementary capabilities of heterogeneous UAVs, their cooperation can provide significant benefits, but it also increases the difficulty of solving the problem [4]. When heterogeneous UAVs perform complex tasks collaboratively, task allocation is one of the critical technologies. This is a discrete combinatorial optimization problem under multiple constraints to establish the mapping relationship between UAVs and tasks. The problem is the global planning of the combat environment,
task constraints, and resources. A reasonable planning method can significantly improve the efficiency of tasks.

Shima et al. [5][6] proposed a cooperative multi-task allocation problem (CMTAP) to solve complex constraints. It assigns multiple UAVs to perform tasks, minimizing the total consumption, which is a combination of task allocation and path planning. CMTAP achieves the general cases of continuous multi-task based on kinematic constraints and cooperations. More importantly, it can get a conflict-free solution among tasks and UAVs.

In practical scenarios, constraints can be complex, such as the protective effect of one target on another, so priority constraints exist among tasks and targets. Target priority constraints are usually ignored by a typical CMTAP, which limits its application. Therefore, this paper extends the CMTAP by fully considering the characteristics of cooperation, introducing heterogeneous UAVs with different capabilities and limited payloads, and establishing priority constraints among multiple targets to propose a cooperative multi-task allocation problem with target priority constraints (CMTAPTPC) to be more realistic. In addition, a novel waitable path coordination (WPC) algorithm is proposed to generate a conflict-free actual flight path that meets the constraints and maximizes the energy saving of the UAVs.

2. PROBLEM FORMULATION

In this section, the task allocation scenario of multiple heterogeneous fixed-wing UAVs performing multi-task against stationary ground targets is described. Then, the work [1] is then extended to create the CMTAPTPC model, which considers the heterogeneity of UAVs and introduces target priority constraints for each target in the graph representation framework.

2.1 Targets and tasks

Let $N_T$ be the number of stationary ground targets and $T = \{T_1, T_2, ..., T_{N_T}\}$ is the set of targets. Additionally, $T_0$ indicates the base where the UAVs need to take off and land.

In actual combat scenarios, targets are usually not isolated. The target priority constraint model can describe the relationship among targets in most typical scenarios. For example, because of combat planning requirements or terrain constraints, $T_i$ must be attacked before $T_j$, or $T_i$ protects $T_j$, and $T_i$ must be destroyed before a reconnaissance mission can be conducted on $T_j$, which $T_i < T_j$ can represent. For each target, the predecessor $P_{T_i}$ and the successor $S_{T_i}$ are used to describe these constraints, as shown in (1).

\[
\begin{align*}
&\{P_{T_i} = \{p_{T_i}^{T_1}, p_{T_i}^{T_2}, ..., p_{T_i}^{T_{N_{P_i}}}\} (i = 0, ..., N_T) \\
&S_{T_i} = \{s_{T_i}^{T_1}, s_{T_i}^{T_2}, ..., s_{T_i}^{T_{N_{S_i}}}\} (i = 0, ..., N_T)
\end{align*}
\]

(1)

where $N_{P_i}$ and $N_{S_i}$ denote the number of predecessor and successor for $T_i$, respectively. For convenience, we define $P_{i}^* = P_{T_i} \cup \{T_i\}, S_{T_i}^* = S_{T_i} \cup \{T_i\}$, so $N_{P_i}^* = N_{P_i} + 1, N_{S_i}^* = N_{S_i} + 1$, and:

\[
\text{sign}(P_{T_i}) = \begin{cases} 
1, & \text{if } P_{T_i} \neq \emptyset \\
0, & \text{if } P_{T_i} = \emptyset
\end{cases}
\]

(2)

$M = \{C, A_s, A_d, V\}$ denotes the set of task types, where $C$ indicates classification task, $A_s$ denotes single attack, $A_d$ denotes double attack performed by different UAVs on the same target, respectively, and $V$ denotes verification task. In addition, these tasks must be executed in the order of $C < A_s < A_d < V$. The set of tasks assigned to each target is indicated by $M_{T_i}^* \subseteq M = \{C, A_s, A_d, V\}, (i = 1, 2, ..., N_T)$.

2.2 UAVs

The different capabilities among UAVs reflect their heterogeneity. Three types of UAVs, combat, surveillance, and munition, are considered in this study. There are also differences in performance among
UAVs of the same type. UAVs with munitions capabilities can only perform attack tasks, and UAVs with surveillance capabilities can perform classification and verification tasks. UAVs with combat capabilities can perform all types of tasks.

Suppose the number of UAVs is $N_u$, and its set is represented by $U = \{U_1^t, U_2^t, ..., U_{N_u}^t\}$, where $t$ represents the three UAV types mentioned above. Dubins car model is adopted as the path planning model for each UAV in this paper. The position of the UAV at any time can be defined by $q = (x, y, \psi)$, where $(x, y)$ and $\psi$ represent the horizontal coordinate and heading angle, respectively. In order to reduce the optimization space, the heading angle must be discretized, which can be expressed by $H = \{\psi_i; \psi_i = 2\pi i/N_\psi, i = 0, 1, ..., N_\psi - 1\}$.

2.3 Graph representation
Using the UAV discrete heading angle established by (3), define the graph:

$$G = (V, E)$$

where

$$V = \left\{ V_{T_0}, V_{T_1}, ..., V_{T_{N_T}} \right\}$$

$$V_{T_i} = \{(T_i, \psi_0), ..., (T_i, \psi_I), ..., (N_\psi - 1)\}$$

$$E = \{ e = (v_{i,j}, w_{i,k}) | v_{i,j} \in V_{T_i}, w_{i,k} \in V_{T_i}^S \}$$

$$i = 0, ..., N_T$$

$$j = 0, ..., N_\psi - 1$$

$$k = 0, ..., N_{S_i}$$

In the directed digraph $G$, $V$ is the set of vertices, the number of which is $|V| = N_V = \sum_{i=0}^{N_T} N_{V_{T_i}} = N_T(N_\psi - 1)$. $E$ denotes the set of edges, $e = (v_{i,j}, w_{i,k})$ means that only after the target $T_i$ is completed can the target in its successor set be executed. Let $V_{T_i}^S$ represents the set of all successor vertices of target $T_i$, the number of which is $N_{S_i}$, so $N_{S_i}^S = N_{V_{T_i}} + N_{S_i}$, and $V_{T_i}^S = V_{T_i} \cup V_{T_i}^S$. Similarly, $V_{T_i}^P$ denotes the set of all predecessors of target $T_i$, the number of which is $N_{P_i}$, and $N_{P_i}^P = N_{V_{T_i}} + N_{P_i}$.

2.4 CMTAPTPC Model
In order to establish the CMTAPTPC model, the target priority constraint is introduced in this study, and soft time windows are also considered to make it more in line with the actual environment. In addition, the model employs stochastic programming with recourse [1] features to improve the robustness of the task allocation scheme. In CMTAPTPC, the objective function consists of the maximum total flight time of the UAV and the expected value of the four penalties, which can be expressed as:

$$\min J = F_0(\mathbf{x}) + E(\mathbf{P})$$

where

$$F_0(\mathbf{x}) = \max_{u \in U} \left\{ \sum_{m=1}^{N_M} \sum_{i=0}^{N_{V_{T_i}}} \sum_{j=0}^{N_{S_i}} \sum_{k=1}^{N_{S_i}^*} \psi_{i,j} u_{m,k} \right\}$$

$$E(\mathbf{P})$$
\[
E(\mathbf{P}) = E(P_1) + E(P_2) + E(P_3) + E(P_4) \\
\begin{align*}
    P_1 &= \sum_{i=1}^{N_T} \lambda_i \max(t_a^T_i - t_c^T_i - \Delta t_{c\alpha}, 0) \\
    P_2 &= \sum_{i=1}^{N_T} \lambda_i^2 \max\left(\left|t_c^T_i - t_a^T_i\right| - \Delta t_{a\alpha}, 0\right) \\
    P_3 &= \sum_{i=1}^{N_T} \lambda_i \text{sign}(\mathbf{P}_T_i) \left[\sum_{j=1}^{N_T} \max\left(t_e^T_j - t_a^T_i - \Delta t_{a\alpha}, 0\right)\right] \\
    P_4 &= \sum_{i=1}^{N_U} \lambda_i^4 \max(t_u - B_u, 0)
\end{align*}
\]  

The objective function is described in (5)-(7). In (5) and (6), \(F_0(\mathbf{x})\) represents the objective function of the deterministic CMTAPTPC model, and \(E(\mathbf{P})\) represents the expected value of punishment calculated under the stochastic programming with recourse model. The calculation method of punishment is described explicitly in (7), in which four punishment types \(P_1, P_2, P_3,\) and \(P_4\) are considered. \(P_1\) and \(P_2\) are the time window punishments of the two types of tasks, respectively, \(P_3\) is the punishment that violates target priority constraint, and \(P_4\) is the punishment for total flight time. \(t_a^T, t_c^T, t_a^T_i,\) and \(t_c^T_i\) represent the execution time of attack, classification, first simultaneous attack and second simultaneous attack, respectively. \(t^T_1\) and \(t^T_2\) denote the execution time of first and last task of \(T_1\), respectively. \(\Delta t_{c\alpha}^T\) \(\Delta t_{a\alpha}^T\) and \(\Delta t_{es}^T\) are predefined time intervals, \(\lambda_1, \lambda_2, \lambda_3\) are the weights of the three types of punishments of target \(T_1\), and \(\lambda_4\) is the weight of total flight time punishment for the \(u\)-th UAV. \(B_u\) and \(t_u\) are the nominal available flight time and the actual flight time, respectively.

3. EXTENDED GRAPH-BASED METHOD FOR SOLVING DEADLOCK

Whether the UAV can perform a task needs to consider its type and remaining resources, which is a problem that must be resolved among heterogeneous UAVs. On the other hand, if the UAV arrives at the target early, it needs to change the flight path to wait for another UAV to complete the task, which increases the risk of deadlock. This section uses the hierarchical graph method [7] to solve the complex deadlock problem caused by heterogeneity. This method is divided into two parts, detecting deadlocks and eliminating deadlocks. If a deadlock is found, it can be transposed until it is unlocked. Fig. 1. shows the flow chart of the proposed method.

![Extended graph-based method flowchart](image)

The task-priority graph (TPG) consists of a task-executing subgraph (TESG) and a task-constraint subgraph (TCSG). The TESG is based on the vehicle-based group (VBG), representing the order execution of the planning solution for each UAV. If a UAV has only one task scheduled, it will not be
represented in the TESG. The TCSG is based on the target-based group (TBG), reflecting the complex priority constraint relationship between targets and tasks. If the TPG has one or more strong connected components (SCC), it indicates a deadlock in the task planning solution, and a conflict-free mission planning solution is generated by performing multiple transpose operations on the TPG to eliminate the SCC.

4. WAITABLE PATH COORDINATION ALGORITHM

In order to coordinate the flight paths among heterogeneous UAVs, the Waiting Path Coordination (WPC) algorithm is proposed in this section. On the one hand, the algorithm can ensure that each UAV arrives at one target in the desired task execution order based on the path elongation method. On the other hand, the algorithm saves UAV energy by allowing UAVs to wait at the base for an appropriate amount of time to avoid excessive path elongation operations.

The WPC algorithm aims to generate a final actual flyable path that satisfies the constraints for each UAV. According to the task planning solution and the difference in the dynamics of each UAV, when a UAV arrived at a target, the previous task for that target may not be completed by another UAV. Therefore, it is necessary to elongate the UAV path that reaches the target prematurely after another UAV completes the previous task. Depending on the topological order, the path elongation method analyzes the flight path of each UAV quantitatively. If the priority constraints are violated, the task execution time of the UAV is changed by adjusting the flight path to generate a new one. The waitable path coordination algorithm is shown in Fig. 2.

| Algorithm: Waitable path coordination algorithm |
|---|
| **Input:** Deadlock-free task planning solution |
| **Output:** Dubins path of each UAV |
| 1. Obtain the topological sorting of the task assignment solution TS. |
| 2. According to the solution, an initial Dubins path is constructed for each UAV. |
| 3. According to the initial Dubins path, the start execution time and end execution time of each task are obtained. |
| 4. for each “target-task” pair $T_i M_j$ ∈ TS, do |
| 5. if $T_i M_j$ is the first task performed by a UAV, then |
| 6. if $T_i M_j$ is the first task of a target, then |
| 7. if $T_i M_j$ is not constrained by target precedence then |
| 8. continue |
| 9. end if |
| 10. end if |
| 11. Calculate the waiting time of the UAV based on the execution time of the previous task of $T_i M_j$. |
| 12. Update the time of all tasks performed by the UAV |
| 13. break |
| 14. end if |
| 15. if The execution time $t_e$ of $T_i M_j$ is reasonable, then |
| 16. continue |
| 17. end if |
| 18. Obtain the UAV executing $T_i M_j$ and the corresponding unreasonable Dubins path segment $D_u$. |
| 19. Obtain the end time of the previous task $t_e$ and the UAV waiting circle time $t_{waitable}$. |
| 20. if $t_e < t_{waitable}$ then |
| 21. if $t_e > t_{waitable}$ then |
| 22. Apply the waiting circle strategy to elongate Dubins path segment $D_u$. |
| 23. else |
| 24. if turn angle less than 180° and $D_u \in \{LSL, RSR, RSI, ISR\}$, then |
| 25. Apply the attached path 2 strategy to elongate Dubins path segment $D_u$. |
| 26. else |
| 27. Apply the attached path 1 strategy to elongate Dubins path segment $D_u$. |
| 28. end if |
| 29. end if |
| 30. Update the time of all tasks performed by the UAV |
| 31. end for |

Figure 2. Waitable path coordination algorithm.
5. SIMULATION
To demonstrate the practicability of the CMTAPTPC model and WPC algorithm, we designed a representative complex simulation scenario, as shown in Table 1. The solution procedure and other parameters are ignored in order to illustrate only the effect of path coordination.

| UAVs Type | Task type | Task priority | Double attack |
|-----------|-----------|---------------|---------------|
| U^c_1     | M_{T_1} = \{C, A_0, V\} | T_1           | T_3           |
| U^m_2     | M_{T_4} = \{C, A_0, V\} | < T_4         | T_6           |
| U^c_3     | M_{T_5} = \{C, A_0, V\} | T_4           | < T_5         |
| U^c_4     | M_{T_6} = \{C, A_0, V\} |               |               |

Table 1. A Complex Simulation Scenario

Fig. 3. shows the time series diagram of the task assignment solution in the above simulation scenario. For U_1, because of T_1 < T_4, it needs to wait for a specific time at the base before taking off. For U_4, it can take off directly to perform the task because it performs the first task of a certain target. U_6 uses the path elongation strategy to wait for another UAV to complete the previous task. According to the execution times of U_2, U_5, and U_1, U_6, it can be seen that they meet the simultaneous attack constraint.

![Time series diagram](image)

6. CONCLUSION
In order to efficiently solve the cooperative multi-task allocation problem, this paper establishes the CMTAPTPC model. In addition, an improved graph method is proposed to solve the complex deadlock caused by heterogeneity. Finally, a novel waitable path coordination algorithm is designed to generate a conflict-free flight path that satisfies the constraints and saves the energy of UAVs. The simulation results demonstrate the effectiveness of the proposed algorithms.

ACKNOWLEDGMENT
The authors greatly appreciate the reviews valuable suggestions and the editors encouragement. This research is supported by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2020JQ-203.

REFERENCES
[1] Jia, Zhenyue, et al. "Cooperative multiple task assignment problem with stochastic velocities and time windows for heterogeneous unmanned aerial vehicles using a genetic algorithm." Aerospace Science and Technology 76 (2018): 112-125.
[2] Schumacher, Corey, et al. "Task allocation for wide area search munitions with variable path length." Proceedings of the 2003 American Control Conference, 2003. Vol. 4. IEEE, 2003.

[3] Darrah, Marjorie, et al. "UAV cooperative task assignments for a SEAD mission using genetic algorithms." AIAA Guidance, Navigation, and Control Conference and Exhibit. 2006.

[4] Zhang, Jun, and Jiahao Xing. "Cooperative task assignment of multi-UAV system." Chinese Journal of Aeronautics (2020).

[5] Shima, Tal, et al. "Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms." Computers & Operations Research 33.11 (2006): 3252-3269.

[6] Edison, Eugene, and Tal Shima. "Integrated task assignment and path optimization for cooperating uninhabited aerial vehicles using genetic algorithms." Computers & Operations Research 38.1 (2011): 340-356.

[7] Deng, Qibo, Jianqiao Yu, and Yuesong Mei. "Deadlock-free consecutive task assignment of multiple heterogeneous unmanned aerial vehicles." Journal of Aircraft 51.2 (2014): 596-605.