Proceedings of the 2020 International Conference on Unmanned Aircraft Systems (ICUAS),
1-4 September, Athens, Greece. DOI:10.1109/ICUAS48674.2020.9213989

Decentralized task allocation for multiple UAVs with task execution uncertainties*

Ruifan Liu, Minguk Seo, Binbin Yan, and Antonios Tsourdos

***Abstract*— This work builds on a robust decentralized task allocation algorithm to address the multiple unmanned aerial vehicle (UAV) surveillance problem under task duration uncertainties. Considering the existing robust task allocation algorithm is computationally intensive and also has no optimality guarantees, this paper proposes a new robust task assignment formulation that reduces the calculation of robust scores and provides a certain theoretical guarantee of optimality. In the proposed method, the Markov model is introduced to describe the impact of uncertain parameters on task rewards and the expected score function is reformulated as the utility function of the states in the Markov model. Through providing the high-precision expected marginal gain of tasks, the task assignment gains a better accumulative score than the state of arts robust algorithms do. Besides, this algorithm is proven to be convergent and could reach a prior optimality guarantee of at least 50%. Numerical Simulations demonstrate the performance improvement of the proposed method compared with basic CBBA, robust extension to CBBA and cost-benefit greedy algorithm.**

I. INTRODUCTION

Effective assignment among UAVs, termed as task allocation, plays a key role in the cooperative control of multiple UAVs[1]. Considering realistic missions are often sophisticated, decomposing it into a series of simple tasks, which are executable for single UAVs or UAV groups, is a general approach in real-world scenarios. Task allocation has a wide range of applications in many networked systems, such as multi-target surveillance[2], grocery delivery[3], search and rescue[4][5], border patrolling to name a few.

Task allocation problems could be considered as combination optimization problems which are shown to be NP-hard. Therefore, obtaining a globally optimal solution is computationally intensive. Under this consideration, many suboptimal approaches are developed, which are generally classified as heuristic approaches and approximate approaches. Approximate approaches are able to provide certain mathematic guarantees of optimality and computational complexity and thus become the most acceptable task allocation method. One typical approximate algorithm is consensus-based auction algorithm (CBBA)[6]. CBBA is an efficient decentralized task allocation algorithm that adopts consensus mechanism with the guarantee of at least 50% optimality. And as the improvement of CBBA, the sampling pre-process and the lazy strategy is integrated to the algorithm in order to improve its scalability[7][8]. These methods show a great performance when addressing the large-scale deterministic task allocation problems, yet they are not applicable to the multi-assignment problem in the uncertain environment, since the discrepancies between the planning model and the actual model would significantly degrade the performance. In an effort to retain the outcome with underlying uncertainties, Ponda[9] then presents the robust extension to CBBA using the integral of expected reward instead. However, analytically computing these robust scores remains cumbersome[9][10].

On the other hand, task allocation methods based on the framework of Markov Decision Process (MDP) have been proposed to deal with the stochastic factors in the real world[11][12][13]. The decentralized MDP (Dec-MDP) is able to figure out the stochastic planning problem in a decentralized manner, which however has been proven to be NEXP-complete even when only two agents are involved[14]. Exact approaches struggle to resolve the assignments[15][16]. Therefore, some literatures work on various ways to reduce the computational cost. One efficient approach is decomposing the Dec-MDP into smaller weakly coupled concurrent processes, solving them individually and merging solutions to form the solution of the original problem. Group aggregated decentralized MDP (GA-Dec-MDP) and Sparse interaction decentralized MDP (SI-Dec-MDP) is both the implementation of this theory[17][18]. Other researches also explore the trait of specific applications to reformulate the problem with hypothesis, such as the assumption of transition independency in [19]. Despite the improvement of performance, these approximate Markov modelling method are often confined with specific applications and none of them is able to model the coupling among tasks and the non-overlapping constraint, which is of importance in the many realistic multi-assignment problems.

Though the existing approaches are hard to solve the problem of interest, they have their own merits. The approximate approach derived from the greedy selection is able to utilize the independence amongst task executions and accomplish the conflict-free assignment efficiently and tractably while MDPs is powerful to formulate the system with uncertainties. Accordingly, a hybrid algorithm merging the strengths of these two method is proposed in [20]. Inspired by [20] and [9], in this paper, we propose a MDP-based robust task allocation algorithm to address the decentralized multi-assignment problem in the stochastic environment where the

*Research supported by China Scholarship Council. (No. 201906290084).

Ruifan Liu is with the School of Astronautics, Northwestern Polytechnical University, 710072, China, and currently visiting with the School of Aerospace, Transport and Manufacturing, Cranfield University, MK43 0AL, UK (e-mail: ruifan.liu@cranfiled.ac.uk).

Minguk Seo is with the School of Aerospace, Transport and Manufacturing, Cranfield University, MK43 0AL, UK (e-mail: minguk.seo@cranfiled.ac.uk).

Binbin Yan is with the School of Astronautics, Northwestern Polytechnical University, 710072, China (yanbinbin@nwpu.edu.cn)

Antonios Tsourdos is with the School of Aerospace, Transport and Manufacturing, Cranfield University, MK43 0AL, UK (e-mail: a.tsourdos@cranfiled.ac.uk).

duration of task execution is uncertain. This proposed decentralized task allocation algorithm has been proven to be convergent and the performance of this algorithm is also verified to be better compared with the basic CBBA, robust extension to CBBA and cost-benefit greedy algorithm.

The rest part of the paper is organized as follows: Section II gives the problem description and its mathematic modelling. Section III presents the detailed explanation of proposed task allocation algorithms and its theoretical analysis. Section IV demonstrates the performance of algorithms through simulation and conducts the comparison with the state of arts approaches. Section V concludes the contribution of the paper and gives a brief plan of future research.

II. Problem formulation

This section gives the description of the mission scenario and the mathematic Markov modelling of this problem, as well as the necessary definitions and basic concepts for the following theoretical analyses.

A. Scenario Description

The multiple-UAV surveillance mission is a time-critical and coupled combinational optimization problem. Supposed that a group of heterogeneous UAVs $i \in A$ are sent to execute a surveillance mission with a list of heterogeneous tasks$j \in \Gamma$, where a non-overlapping task scheme need to be found to maximize the reward function, formulated as (1):

$$\max_{\boldsymbol{x}} \sum_{i=1}^{|A|} \sum_{j=1}^{|\Gamma|} c_{ij}(\boldsymbol{x}_i, \boldsymbol{\theta}) \tag{1}$$

subject to

$$\begin{cases} \sum_{i=1}^{|A|} x_{ij} \le 1 & \forall j \in \Gamma \\ \sum_{j=1}^{|\Gamma|} x_{ij} \le L_i & \forall i \in A \\ x_{ij} \in \{0,1\} & \forall (i,j) \in A \times \Gamma \end{cases} \tag{2}$$

where $x_{ij}$is a binary decision variable indicating whether task j is allocated to UAVi, c_{ij} is the marginal score of task j given the tasks that are already allocated to UAVi, shown as $\boldsymbol{x}_i$, θ_{ij} is the uncertain parameter related to the score calculation, representing the task duration time in this paper. The first equation of constraints denotes the non-overlapping constraint, i.e. each task is assigned to no more than one UAV. And the second one denotes that the cumulative number of tasks for each UAV is no more than its capabilityL_i.

In order to model time-critical scenario, the reward function is formulated with a time-discounted portion, established as (3):

$$J = \sum_{i=1}^{|A|} R_i = \sum_{i=1}^{|A|} \sum_{j=1}^{|\Gamma_i|} m_{ij} v_j e^{-\lambda \tau_{ij}(\boldsymbol{\theta})} \tag{3}$$

where $\Gamma_i \subseteq \Gamma$ is the list of tasks selected by UAVi, m_{ij}andv_j represent the fitness factor between the task j and UAVi and the importance factor of task j respectively, λ is the time-discount index and τ_{ij}means the execute time of task j.

This paper concerns the task allocation problem under task execution uncertainties. Most of the former task allocation method assumes that the execution of the task is performed for a short period of time or completed instantaneously. However, considering the complexity of the real environment, the task duration of each UAV-task pair differs a lot due to their own characteristic and load capacity, and usually behaves as a random process. Thus this paper focus on how to reduce the impact of the uncertain task duration and assumes that the time on the journey is ignorable.

In specific multi-UAV task allocation problems, the uncertain duration often have their own distribution, which can be obtained from historical data, surveys or theoretical analysis[10]. While the true value of θ is unknown, it is assumed that a likelihood model of the uncertainty parameter is known beforehand, obeying a statistical probability distribution:

$$\theta_{ij} \sim \Pr(\theta_{ij}) \tag{4}$$

B. Markov Modelling

Here assumed that each UAV has local full observation with the environment and states of itself. Identical with centralize MDP, decentralized MDP also has five components.

1) Decision epochs: the decision making points correspond to the number of targets N_t . That's to say, allocation is only made for one task in every decision epoch.

2) State: State is factored into the individual state of each UAV: $s_i = (\tau_i, \Gamma^i_{ava})$, where τ_i denotes the time that have been consumed by UAV i, Γ^i_{ava} is the set of available tasks for UAVi. It is noted that when Γ^i_{ava} becomes $\emptyset$, UAVi reaches the final stateS_f.

3) Action: action indicates the next task to be executed for UAVi under the current state, presented by selecting a task from the available task set, $a_i = j, j \in \Gamma^i_{ava}$.

4) Reward Function: The reward function is the return value obtained from the environment after the system performs the action. We define the reward as an immediate gain by executing the next selected task. Hence, the reward function is presented as

$$r(s_i, a_i) = m_{ij} v_j e^{-\lambda \tau_j} \tag{5}$$

5) Transition Probability: $Pr(s'_i|s_i, a_i)$ represents the probability that the current state s_i is converted to the next state$s'_i$by taking actiona_i. For the task allocation problem discussed here, the task duration of each task-UAV pair decides the next state of the system and the probability of duration time defines the transition probability as well:

$$Pr(s'_i|s_i, a_i) = Pr(\theta_{ij}|i, j)$$

As assumed in the last section, that actual duration follows a normal distribution with alternative variance according to the type of tasks and UAVs.

During the allocation process, the next state of UAV i is updated as:

$$s_i' = \begin{Bmatrix} \tau_i + \theta_{ij} \\ \Gamma_{ava}^i \backslash \{j\} \end{Bmatrix}$$

C. Preliminary

This part gives some necessary preliminary and concept to development and analysis of the proposed task allocation method.

Definition 1: (Submodularity) let N be a finite set. A real-valued set function $f: 2^{\mathrm{N}} \rightarrow R$ is submodular if, for all $X, Y \subseteq N$,

$$f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y)$$

Equivalently, for all $A \subseteq B \subseteq N$ and$u \in N \backslash B$,

$$f(A \cup \{u\}) - f(A) \geq f(B \cup \{u\}) - f(B) \qquad (6)$$

Definition 2: (Monotonicity) A set function $f: 2^{\mathrm{N}} \rightarrow R$ is monotone if, for every$A \subseteq B \subseteq N$,

$$f(B) \geq f(A)$$

Definition 3: (Matroid) A matroid is a pair $M = (N, I)$ where N is a finite set and $I \subseteq 2^N$ is a collection of independent sets, satisfying:

- $\emptyset \in I$
- $A \subseteq B, B \in I \Rightarrow A \in I$
- $A, B \in I, |A| < |B| \Rightarrow \exists b \in B \backslash A$ such that $A \cup \{b\} \in I$.

We are interested in a ground set that is partitioned as$N = N_1 \cup N_2 \cup \cdots \cup N_k$. The collection of subsets, $I = \{I \subseteq N: \forall i, |P \cup N_i| \leq 1\}$ forms a matroid called a partition matroid. According to the problem scenario, only one task-UAV pair from the task-UAV pairs is allowed to be selected. If all task-UAV pairs are considered as a ground set (i.e. $N: \Gamma \times A$) and each task-UAV pair as an element of the ground set. Thus the task allocation problem modelled in this paper could be handled as an maximization problem subject to a partition matroid constraint[8].

Monotone submodular function has good mathematical characters, with which difficulty can be approximated efficiently with a strong quality guarantee[21]. Specially, a greedy algorithm that incrementally choose elements by maximizing marginal utility provides solution with at least $(1 - 1/e)$ optimality and 1/2 optimality if there is a matroid constraint[21].

III. Task Allocation Method

This section presents the MDP-based decentralized task allocation algorithm to handle the uncertainties lying under the problem, the basic idea coming from the robust extension to CBBA[9].

A. Robust Extension to CBBA

In [9], Ponda develops an extension to CBBA, improving the original CBBA with the robust model of expected reward function. The basic CBBA is a decentralized task allocation algorithm, consisting of two separated phases: the bundle construction phase where a bundle of tasks is greedily chosen by each UAV, and the task consensus phase where the conflicts of tasks are resolved through communicating and negotiating with neighbours. The algorithm iterates between these two phases until all tasks are settled in a consistent manner.

For the purpose of preserving the performance under uncertain parameters, the robust strategy models uncertain parameters into the reward function and optimizes the expected value considering the uncertainty distribution. The objective function is proposed as:

$$\max J_i = E_\theta \left\{ \sum_{j=1}^{|\Gamma|} c_{ij}(\boldsymbol{x}_i, \boldsymbol{\theta}) \right\} \qquad (7)$$

The uncertainty of task duration affects the reward score of each tasks, as well as the task orders for each UAV's task bundle since the coupling between tasks. Thus when constructing the task bundle for each UAV, every possible task order must be taken into account. Thus the score of task bundle is calculated as:

$$\begin{aligned} J_i &= \max_{\mathbf{p}_i} J_{\mathbf{p}_i} = \max_{\mathbf{p}_i} E_{\boldsymbol{\theta}} \left\{ \sum_{j=1}^{|\Gamma|} c_{ij} \left(\tau_{ij}(\mathbf{p}_i, \boldsymbol{\theta}) \right) x_{ij} \right\} \\ &= \max_{\mathbf{p}_i} \int_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \left\{ \sum_{j=1}^{|\Gamma|} c_{ij} \left(\tau_{ij}(\mathbf{p}_i, \boldsymbol{\theta}) \right) x_{ij} \right\} P(\boldsymbol{\theta}) \, d\boldsymbol{\theta} \end{aligned}$$

where $\mathbf{p}_i$ indicates the task sequence, following which the task will be executed one by one.

By means of extracting the uncertainty parameters and optimizing the priori-estimated reward score, this extension could indeed enhance the robustness of CBBA algorithm in the uncertain environment. However, analytically computing this robust score is very difficult due to the integral operation of uncertainty parameters and the numerous permutations of the task order. For this reason, some researches adopt a sampling process to approximate these calculations and assume that the order of existing tasks is fixed in order to maintain computational tractability[9], while the sampling technique might violate the *diminishing marginal gain* (DMG) condition of the algorithm convergence.

Therefore, the next section proposes a theoretically analyzable and computationally tractable approach to acquire the robust score, which employs the theory of MDP.

B. Task Allocation Algorithm

The proposed algorithm has the similar framework with the basic CBBA, iterating between the two phases, shown as Fig. 1. The main upgradation of the method is located in the bundle construction phase and the calculation of marginal value.

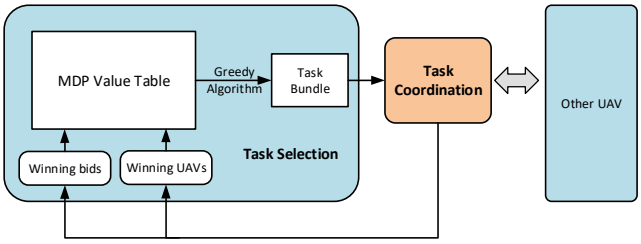

Figure 1. The Framework of MDP-based CBBA

As shown in the Fig. 1, the task bundle for each UAV is constructed by greedily selecting the task with the largest MDP value. The value for every state in the Markov model is calculated by the policy iteration approach before allocation starts. Note that if the algorithm is implemented on a more time-critical or scalable application, learning method could be employed to estimate the expected value for every state[22][23].

1. Calculation of Marginal Gain

In the proposed method, the marginal gain of tasks is calculated by the utility value for states in MDPs, which indicates the expected accumulative reward in the future. Thus the objective function of robust CBBA could be equivalent to the value function under a specific state. According to the definition of the value function for the finite-horizon MDP,

$$V_\pi(s_0) = E\left[\sum_{i=1}^{H} R(s_i, \pi(s_i), s_{i+1})\right],$$

the maximum objective function described as (7) could be converted to

$$\max J_i(\Gamma_i) \Rightarrow V_\pi^*(s), s = \{0, \Gamma_i\} \tag{8}$$

where $s = \{0, \Gamma_i\}$ means that the initial time is 0 and the available task set is Γ_i. Thus, through reformulating the objective function into the value function of MDP, the maximum objective could be acquired by solving the optimal utility of MDP. And the marginal value of task j could be further calculated by the minus of values under two states:

$$c_{ij} = J^*(\Gamma_i \cup \{j\}) - J^*(\Gamma_i) = V_\pi^*(s_1) - V_\pi^*(s_2)$$

where $s_1 = (0, \Gamma_i \cup \{j\})$, $s_2 = (0, \Gamma_i)$.

The pseudocode of the bundle construction of the proposed decentralized task allocation method is presented as Algorithm 1.

TABLE I. MDP-BASED CBBA BUNDLE CONSTRUCTION

Algorithm1: MDP-based CBBA Bundle Construction for UAV *i*

```
BUILD BUNDLE (z_i, y_i, b_i)
while h not empty
    for j ∈ Γ \ b_i
        s_1 = (0, b_i ∪ {j}), s_2 = (0, b_i)
        c_ij = V_π*(s_1) − V_π*(s_2)
        h_ij = 𝕀(c_ij > y_ij)
    end for
    j* ← argmax_{j∉b_i} c_ij · h_ij
    b_i ← (b_i ∪ {j*})
    y_ij* ← c_ij*
    z_ij* ← i
end while
```

Another advantage of this MDP-formulated method is that all possible orders of tasks in the bundle will be automatically considered when calculating the score function. Most of existing task allocation algorithms assume that the order of selected tasks does not change when new tasks are inserted into the sequence in order to reduce computations. This assumption actually ignores the impact of new tasks on the optimal execution order and thus decrease its performance, violating the guarantee of the optimality. The proposed method calculates the marginal gain by the utility of the task bundle, where every permutation of the task bundle is taken into account, truly holding the properties of greedy algorithms.

C. Theoretical Analysis

The proposed task allocation method merges the MDP theory into the bundle construction phase of the robust CBBA maintaining the convergence and the guarantee optimality of the algorithm. The relevant theoretical proofs are provided in this section, where the submodularity of the static reward function is firstly proven as a prerequisite and then the convergence and optimality analysis are presented based on the submodular deterministic reward.

1. Submodularity Proof of Deterministic Reward

The deterministic reward function is defined by (3), where the parameter θ is assumed to be deterministic. Since the sum of submodular functions is still submodular, the proof is given for the reward function of each UAV:

$$R(\Gamma_i) = \sum_{j=1}^{|\Gamma_i|} m_j v_j e^{-\lambda \tau_j(\boldsymbol{p_i})} \tag{9}$$

Lemma1: For reward function(9), if p_i^* is the optimal order of tasks in the task setΓ_i, for any task j_a in p_i^*, it would satisfies

$$\frac{m_{j_a} v_{j_a}}{1 - e^{-\lambda t_{j_a}}} \geq \frac{m_j v_j}{1 - e^{-\lambda t_j}} \ \forall j \ \text{after} j_a \ \text{in} \ p_i^* \tag{10}$$

where t_j represents the task duration of task$j$.

Proof: Suppose that $p_i^*$is the optimal order but not satisfies the condition, exchange every adjacent task pairs that do not satisfy the inequality until every tasks follow the inequality. The accumulative reward changes while shifting the order of tasks, which could be calculated as the sum of every adjacent exchanges, presented as follows:

$$\begin{aligned}\Delta R &= \sum \big(r(j_2, j_1) - r(j_1, j_2)\big) \\ &= \sum \Big[m_{j_2} v_{j_2} \left(e^{-\lambda \tau_0} - e^{-\lambda(\tau_0 + t_{j_1})}\right) \\ &\quad - m_{j_1} v_{j_1} \left(e^{-\lambda \tau_0} - e^{-\lambda(\tau_0 + t_{j_2})}\right)\Big]\end{aligned}$$

It has been assumed that (j_1, j_2) does not satisfy the inequality thus:

$$\frac{m_{j_1} v_{j_1}}{1 - e^{-\lambda t_{j_1}}} < \frac{m_{j_2} v_{j_2}}{1 - e^{-\lambda t_{j_2}}}$$

Combining these two equations, we can conclude that $\Delta R > 0$, which is conflict with the assumption that p_i^* is the optimal order. Hence, the *Lemma1* has been proven.■

Theorem1: the reward function of task allocation problem defined as (9) is submodular.

Note that the reward function could not be simply proven to be submodular even if the terms within the summation symbol is submodular, since the number of terms of the weighted sum changes with different task sets.

Proof: With the knowledge of *Lemma1*, every pair of tasks in the optimal order must obey the inequality(9), the insertion of new tasks will not affect the order of existing tasks in the task sequence, and also the insert position might move backwards while more tasks are in the set. Thus for the ease of analysis, we can decompose the task sequence according to the insert position of the new task.

Given two task setsΓ_m and Γ_n ,$\Gamma_m \subseteq \Gamma_n$. The marginal gain of set Γ_m is

$$\begin{aligned} &R(\Gamma_m \cup \{j\}) - R(\Gamma_m) \\ &\quad = \left[R\left(\Gamma_m^{k_1-}\right) + m_j v_j e^{-\lambda \tau_{k_1}} + R'\left(\Gamma_m^{k_1+}\right)\right] \\ &\quad - \left[R\left(\Gamma_m^{k_1-}\right) + R\left(\Gamma_m^{k_1+}\right)\right] \\ &\quad = R'\left(\Gamma_m^{k_1+}\right) - R\left(\Gamma_m^{k_1+}\right) + m_j v_j e^{-\lambda \tau_{k_1}} \\ &= e^{-\lambda \tau_{k_1}} \left(m_j v_j - \sum\nolimits_{i \in \Gamma_m^{k_1+}} m_i v_i \left(1 - e^{-\lambda t_j}\right) \right) \end{aligned}$$

where$k_1$is the insertion position for task j, t_j is the duration time of task j, $\Gamma_m^{k_1-}$and $\Gamma_m^{k_1+}$indicate the task sequence before the insertion and after insertion respectively, and τ_{k_1}represents the execution time of task j, which is calculated as

$$\tau_{k_1} = \sum\nolimits_{i \in \Gamma_m^{k_1-}} t_i$$

Similarly, the marginal gain of set Γ_n is

$$\begin{aligned} &R(\Gamma_n \cup \{j\}) - R(\Gamma_n) \\ &\quad = R\left(\Gamma_n^{k_2+}\right) - R'\left(\Gamma_n^{k_2+}\right) + m_j v_j e^{-\lambda \tau_{k_2}} \\ &= e^{-\lambda \tau_{k_2}} \left(m_j v_j - \sum\nolimits_{i \in \Gamma_n^{k_2+}} m_i v_i \left(1 - e^{-\lambda t_j}\right) \right) \end{aligned}$$

Let us now compare these two equations. According to *Lemma1* , it is easy to know that $\Gamma_m^{k_1+} \subseteq \Gamma_n^{k_2+}$, as well as $\tau_{k_2} \geq \tau_{k_1}$. Thus

$$e^{-\lambda \tau_{k_1}} \geq e^{-\lambda \tau_{k_2}},$$

$$\sum\nolimits_{i \in \Gamma_m^{k_1+}} m_i v_i \leq \sum\nolimits_{i \in \Gamma_n^{k_2+}} m_i v_i,$$

the decreasing value with task set $\Gamma_n^{k_2+}$ is not greater than the one with set$\Gamma_m^{k_1+}$, and the reward of new task inserted in position k_2 is also smaller than in positionk_1. Hence, we could conclude

$$R(\Gamma_m \cup \{j\}) - R(\Gamma_m) \geq R(\Gamma_n \cup \{j\}) - R(\Gamma_n)$$

Theorem1 holds. ■

2. Convergence Analysis

The convergence condition of the MDP-based CBBA ,same with the basic CBBA, is the score function must satisfy DMG, which have the same denotation with submodularity[6]. The submodularity proof of robust reward function is given by *Theorem2.*

Theorem2: if the objective function of the task allocation problem with deterministic parameters is submodular, the MDP value-presented robust objective function is also submodular.

Proof: For ease of analysis, we give a new presentation of the MDP value function:

$$V(t, \Gamma_{ava}) \Leftarrow V_\pi^*(s), s = \{t, \Gamma_{ava}\} \tag{6}$$

where t is the time already consumed by UAV, $\Gamma_{ava}$is the available task set.

Now let us consider two available task sets Γ_m and Γ_n ,$\Gamma_m \subseteq \Gamma_n$. In order to prove the submodularity of the value function defined as (8), the task set $\Gamma_n \cup \{j\}$ is factored into 3 parts $\left[\Gamma_m, \Gamma_{n\backslash m}, \{j\}\right]$ and allocated one by one. Since the value for MDP states is only relevant with the entries in the task set, thus the order of allocation have no impact to the transition and accumulative reward. Thus based on the Bellman equations for MDPs, the marginal gain with the prior of task set Γ_n is presented as:

$$\begin{aligned} &V(0, \Gamma_n \cup j) - V(0, \Gamma_n) \\ &= \int_{\boldsymbol{\theta}_{\Gamma_m}} Pr(\boldsymbol{\theta}_{\Gamma_m}) [R(\boldsymbol{\theta}_{\Gamma_m}) + V(\boldsymbol{\theta}_{\Gamma_m}, \Gamma_{n\backslash m} \cup j)] d\boldsymbol{\theta}_{\Gamma_m} \\ &- \int_{\boldsymbol{\theta}_{\Gamma_m}} Pr(\boldsymbol{\theta}_{\Gamma_m}) [R(\boldsymbol{\theta}_{\Gamma_m}) + V(\boldsymbol{\theta}_{\Gamma_m}, \Gamma_{n\backslash m})] \, d\boldsymbol{\theta}_{\Gamma_m} \\ &= \int_{\boldsymbol{\theta}_{\Gamma_m}} Pr(\boldsymbol{\theta}_{\Gamma_m}) \Bigg\{ R(\boldsymbol{\theta}_{\Gamma_m}) \\ &+ \int_{\boldsymbol{\theta}_{\Gamma_{n\backslash m}}} Pr\left(\boldsymbol{\theta}_{\Gamma_{n\backslash m}} \middle| \boldsymbol{\theta}_{\Gamma_m}\right) \left[R\left(\boldsymbol{\theta}_{\Gamma_{n\backslash m}}\right) + r(\boldsymbol{\theta}_{\Gamma_n}, j)\right] d\boldsymbol{\theta}_{\Gamma_{n\backslash m}} \Bigg\} d\boldsymbol{\theta}_{\Gamma_m} \\ &- \int_{\boldsymbol{\theta}_{\Gamma_m}} Pr(\boldsymbol{\theta}_{\Gamma_m}) \Bigg\{ R(\boldsymbol{\theta}_{\Gamma_m}) \\ &+ \int_{\boldsymbol{\theta}_{\Gamma_{n\backslash m}}} Pr\left(\boldsymbol{\theta}_{\Gamma_{n\backslash m}} \middle| \boldsymbol{\theta}_{\Gamma_m}\right) R\left(\boldsymbol{\theta}_{\Gamma_{n\backslash m}}\right) d\boldsymbol{\theta}_{\Gamma_{n\backslash m}} \Bigg\} d\boldsymbol{\theta}_{\Gamma_m} \\ &= \int_{\boldsymbol{\theta}_{\Gamma_m}} Pr(\boldsymbol{\theta}_{\Gamma_m}) \int_{\boldsymbol{\theta}_{\Gamma_{n\backslash m}}} Pr\left(\boldsymbol{\theta}_{\Gamma_{n\backslash m}} \middle| \boldsymbol{\theta}_{\Gamma_m}\right) r(\boldsymbol{\theta}_{\Gamma_n}, j) d\boldsymbol{\theta}_{\Gamma_{n\backslash m}} \, d\boldsymbol{\theta}_{\Gamma_m} \end{aligned}$$

where

$$Pr(\boldsymbol{\theta}_{\Gamma_m}) = \prod\nolimits_{j \in \Gamma_m} Pr(\theta_j),$$

$$R(\boldsymbol{\theta}_{\Gamma_m}) = \sum\nolimits_{j \in \Gamma_m} r(\boldsymbol{\theta}_{\Gamma_m}, j)$$

Similarly, the marginal gain with task set Γ_m is presented as:

$$\begin{aligned} &V(0, \Gamma_m \cup j) - V(0, \Gamma_m) \\ &= \int_{\boldsymbol{\theta}_{\Gamma_m}} Pr(\boldsymbol{\theta}_{\Gamma_m}) [R(\boldsymbol{\theta}_{\Gamma_m}) + r(\boldsymbol{\theta}_{\Gamma_m}, j)] d\boldsymbol{\theta}_{\Gamma_m} \\ &- \int_{\boldsymbol{\theta}_{\Gamma_m}} Pr(\boldsymbol{\theta}_{\Gamma_m}) [R(\boldsymbol{\theta}_{\Gamma_m})] \, d\boldsymbol{\theta}_{\Gamma_m} \\ &= \int_{\boldsymbol{\theta}_{\Gamma_m}} Pr(\boldsymbol{\theta}_{\Gamma_m}) r(\boldsymbol{\theta}_{\Gamma_m}, j) d\boldsymbol{\theta}_{\Gamma_m} \end{aligned}$$

Since the deterministic reward function r is assumed to be submodular, thus: $r(\boldsymbol{\theta}_{\Gamma_n}, j) \leq r(\boldsymbol{\theta}_{\Gamma_m}, j)$. With the knowledge that $\int_{\boldsymbol{\theta}_{\Gamma_{n\backslash m}}} Pr\left(\boldsymbol{\theta}_{\Gamma_{n\backslash m}} \middle| \boldsymbol{\theta}_{\Gamma_m}\right) d\boldsymbol{\theta}_{\Gamma_{n\backslash m}} = 1$, we conclude

$$V(0, \Gamma_n \cup j) - V(0, \Gamma_n) \leq V(0, \Gamma_m \cup j) - V(0, \Gamma_m)$$

■

3. Optimality Analysis

As presented above, the robust score function is proven to be submodular, and obviously the score function is also monotone since the accumulative reward is not going to decrease even there are more tasks in the bundle. Besides, the non-overlapping constraint could be considered as a partition matroid where all task-UAV pairs are defined as a ground set $N: \Gamma \times A$ and the collection of subsets, which is limited to at most one element from the same subset.

According to [24], for the problem maximizing a monotone submodular function subject to a matroid constraint, the greedy algorithm is guaranteed to produce a solution that is bigger than 50% of the optimal solution. The CBBA holds the identical solution with the greedy algorithm when the reward function is DMG. Accordingly, the MDP-based CBBA retains the guarantee of at 50% optimality, since the robust score function has been proven to be submodular through *Theorem2.*

Note that the optimality guarantee is refer to the optimal expected reward rather than the actual execution reward as the uncertain parameter would not be obtained unless the task is really executed.

To summarise, the proposed MDP-based task allocation algorithm, as a robust extension of CBBA, reformulates the basic reward with the expected value of MDPs, thus enhancing the robustness under uncertain task durations and meanwhile maintaining the DMG convergence condition of the algorithm, as well as the optimal guarantee of 50%.

IV. Simulation Result

In this section, the proposed MDP-based task allocation algorithm is testified with a group of UAVs executing surveillance mission. The simulation result is compared among the proposed MDP-based robust CBBA, basic version of CBBA, sampling robust CBBA[9] and cost-benefit greedy algorithm[24]. Samples $N = 10000$ are used in the sampling robust CBBA to calculate the expected value and the wrapping method is also adopted to keep convergence. The cost-benefit greedy algorithm, which usually performs well with the cost constraint, takes cost into account and iteratively adds

$$\Gamma_{i+1} = \Gamma_i \cup \left\{ \underset{j \in \Gamma \setminus \Gamma_i : t_j \leq d - \tau(\Gamma_i)}{\operatorname{argmax}} \frac{\Delta R(j|\Gamma_i)}{t_j} \right\}$$

i.e. the element$j$ that maximizes the benefit cost (refer to time consumption in this paper) ratio among all elements still affordable with the remaining budget.

Suppose that there are 10 heterogeneous tasks and 2 heterogeneous UAVs of different types equipped with different sensors are sent to these tasks for surveillance mission. Each task has the different priority factor and different match factors with each UAV, which is all listed in TABLE II.

The distribution of task duration for every task-UAV pair is assumed to follow the normal distribution with the same standard deviation, set as 1.0, and the mean value of each normal distribution depends on each UAV-task pair, which is also listed in TABLE II. The time discount index is set as$\lambda = 0.1$.

TABLE II. Task Features

Task id	Priority Factor	Fitness Factor		Mean of Task Duration	
		UAV1	UAV2	UAV1	UAV2
1	0.553	0.671	0.944	1.754	1.228
2	0.752	0.839	0.688	1.915	1.399
3	0.621	0.588	0.575	1.217	1.247
4	0.928	0.934	0.889	1.266	1.505
5	0.846	0.859	0.722	1.826	1.484
6	0.985	0.999	0.552	1.283	1.745
7	0.601	0.705	0.958	1.949	1.376
8	0.741	0.601	0.995	1.529	1.060
9	0.699	0.522	0.770	1.929	1.771
10	0.729	0.922	0.532	1.940	1.919

Based on the above setting, the task assignment results and the planning scores are shown in the Table III. In order to verify the performance on real stochastic environment, the task execution simulation is conducted for 100 rounds, where the task duration time is randomly generated with the given normal distribution, whose result is also given in Table III.

TABLE III. Task Assignment Result

Algorithms	Task assignment result		Planning Score	Actual Score
CBBA	UAV1	[6,1,4]	4.6866	4.1824
	UAV2	[8,3,5,7,9]		
CT-Greedy	UAV1	[6,1,4,2]	4.5697	4.2053
	UAV2	[8,3,5,7,9]		
Sampling Robust CBBA	UAV1	[6,1,2,10]	4.3873	4.3663
	UAV2	[8,3,5,7,4,9]		
MDP-based Robust CBBA	UAV1	[6,2,10]	4.5058	4.4972
	UAV2	[8,5,4,7,3,1,9]		

In terms of the different task number and randomly generated parameters, the Monte Carlo simulation is also conducted amongst proposed MDP-based CBBA, CBBA and cost-benefit greedy (CB-Greedy). We run 20 rounds of Monte Carlo simulations and 100 rounds of verification simulations in stochastic environment for every Monte Carlo simulation. In the beginning of every Monte Carlo simulation, the priority factor and fitness factor of each UAV-task pair are both generated following a uniformly random distribution: $m_{aj} \sim U(0.5,1.0)$, $v_j \sim U(0.5,1.0)$. The accumulative score and the compute time for these 4 algorithms are respectively depicted as Fig. 2 and Fig. 3.

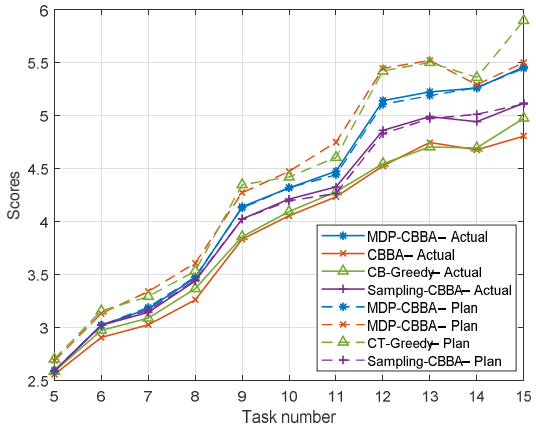

Figure 2. Total task scores: Plan vs. Actual

As shown in Fig. 2, the two robust CBBA algorithms achieve higher overall rewards in actual mission execution and deviate much littler from the planning results than the deterministic algorithms do. This is because the robust strategy captures the influence of the uncertainties on the reward function and the task assignment. Moreover, the MDP-based robust CBBA gains a better performance than the sampling robust CBBA does, for the reason that the Markov modelling considers the uncertainty more comprehensively than the limited samples does and avoids the assumption of the fixed order which could also reduce the performance.

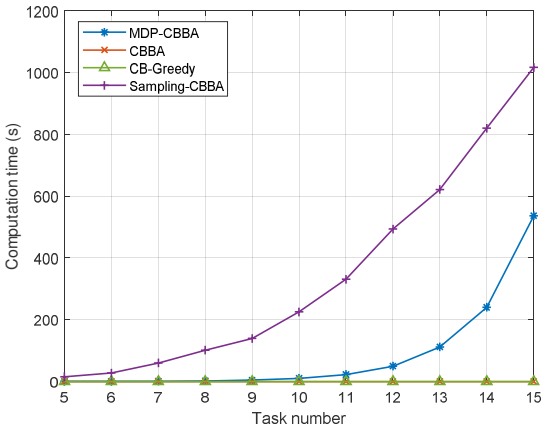

Figure 3. Total running time

As for the running time shown in Fig. 3, with the fact that the MDP-based robust CBBA consumes less time than the sampling robust algorithm does, the computational cost of both methods is still unaffordable for large-scale task allocation problems. Since the most of the computation time in MDP-based robust CBBA is spent on constructing the transition model of MDPs, a more computational efficient method to calculate the utility still needs to be developed in the future.

V. Conclusion

This paper presents a decentralized task allocation algorithm addressing task allocation problems with execution time uncertainties. Through modelling the uncertainties into MDPs, the proposed MDP-based robust task allocation algorithm acquires a more precise expectation of marginal value than the state of arts algorithms does, leading to a more robust allocation in the stochastic environment. Moreover, this algorithm is proven to retain the convergence of the decentralized task consensus and could reach the optimality guarantee of at least 50% referring to the expected reward. The performance of the proposed method is verified by Monte Carlo simulation and compared with the deterministic CBBA algorithm, sampling robust CBBA and cost-benefit greedy algorithm.

Future works are planned to be carried out from two aspects: explore a more efficiency way like deep learning method to estimate the utility value of large-size MDPs since the dimensions explode when the number of tasks and UAVs increases; integrate the information of other UAVs into dependent MDPs in order to reduce the communication burden brought by the task negotiation in the proposed method.

References

[1] Brutschy, A., Pini, G., Pinciroli, C., Birattari, M. and Dorigo, M. (2014) 'Self-organized task allocation to sequentially interdependent tasks in swarm robotics', *Autonomous Agents and Multi-Agent Systems*, 28(1), pp. 101–125. doi: 10.1007/s10458-012-9212-y.

[2] Shakhatreh, H., Sawalmeh, A. H., Al-Fuqaha, A., Dou, Z., Almaita, E., Khalil, I., Othman, N. S., Khreishah, A. and Guizani, M. (2019) 'Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges', *IEEE Access*. IEEE, 7, pp. 48572–48634. doi: 10.1109/ACCESS.2019.2909530.

[3] Akkarajitsakul, K., Hossain, E. and Niyato, D. (2013) 'Coalition-based cooperative packet delivery under uncertainty: A dynamic bayesian coalitional game', *IEEE Transactions on Mobile Computing*. IEEE, 12(2), pp. 371–385. doi: 10.1109/TMC.2011.251.

[4] Scherer, J., Yahyanejad, S., Hayat, S., Yanmaz, E., Vukadinovic, V., Andre, T., Bettstetter, C., Rinner, B., Khan, A. and Hellwagner, H. (2015) 'An autonomous multi-UAV system for search and rescue', *DroNet 2015 - Proceedings of the 2015 Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, pp. 33–38. doi: 10.1145/2750675.2750683.

[5] Zhao, W., Meng, Q. and Chung, P. W. H. (2016) 'A Heuristic Distributed Task Allocation Method for Multivehicle Multitask Problems and Its Application to Search and Rescue Scenario', *IEEE Transactions on Cybernetics*. IEEE, 46(4), pp. 902–915. doi: 10.1109/TCYB.2015.2418052.

[6] Choi, H. L., Brunet, L. and How, J. P. (2009) 'Consensus-based decentralized auctions for robust task allocation', *IEEE Transactions on Robotics*, 25(4), pp. 912–926. doi: 10.1109/TRO.2009.2022423.

[7] Shin, H.-S., Li, T. and Segui-Gasco, P. (2019) 'Sample Greedy Based Task Allocation for Multiple Robot Systems', pp. 1–25. Available at: http://arxiv.org/abs/1901.03258.

[8] Li, T., Shin, H.-S. and Tsourdos, A. (2019) 'Efficient Decentralized Task Allocation for UAV Swarms in Multi-target Surveillance Missions', *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, pp. 61–68. doi: 10.1109/icuas.2019.8798293.

[9] Ponda, S. S. (2012) 'Robust Distributed Planning Strategies for Autonomous Multi-Agent Teams', *ProQuest Dissertations and Theses*, 0828990. doi: 1721.1/77100.

[10] Wasik, D., Mulchandani, A. and Yates, M. V. (2018) 'An efficient sampling-based algorithms using active learning and manifold learning for multiple unmanned aerial vehicle task allocation under uncertainty', *Sensors (Switzerland)*, 18(8). doi: 10.3390/s18082645.

[11] Mausam and Kolobov, A. (2012) *Planning with markov decision processes: An AI perspective*, *Synthesis Lectures on Artificial Intelligence and Machine Learning*. doi: 10.2200/S00426ED1V01Y201206AIM017.

[12] Kumar, R. R., Varakantham, P. and Kumar, A. (2017) 'Decentralized planning in stochastic environments with submodular rewards', *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, pp. 3021–3028.

[13] Agrawal, P., Varakantham, P. and Yeoh, W. (2018) 'Decentralized

planning for non-dedicated agent teams with submodular rewards in uncertain environments', *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, 2, pp. 958–967.

[14] Bernstein, D. S., Givan, R., Immerman, N. and Zilberstein, S. (2002) 'The complexity of decentralized control of Markov decision processes', *Mathematics of Operations Research*, 27(4), pp. 819–840. doi: 10.1287/moor.27.4.819.297.

[15] Kim, I. and Morrison, J. R. (2018) 'Learning Based Framework for Joint Task Allocation and System Design in Stochastic Multi-UAV Systems', *2018 International Conference on Unmanned Aircraft Systems, ICUAS 2018*. IEEE, pp. 324–334. doi: 10.1109/ICUAS.2018.8453318.

[16] Kim, M. and Morrison, J. R. (2019) 'On systems of UAVs for persistent security presence: A generic network representation, MDP formulation and heuristics for task allocation', *2019 International Conference on Unmanned Aircraft Systems, ICUAS 2019*. IEEE, pp. 238–245. doi: 10.1109/ICUAS.2019.8797863.

[17] Redding, J. D. (2011) 'Approximate Multi-Agent Planning in Dynamic and Uncertain Environments'.

[18] Melo, F. S. and Veloso, M. (2011) 'Decentralized MDPs with sparse interactions', *Artificial Intelligence*. Elsevier B.V., 175(11), pp. 1757–1789. doi: 10.1016/j.artint.2011.05.001.

[19] Becker, R., Zilberstein, S., Lesser, V. and Goldman, C. V. (2004) 'Solving transition independent decentralized Markov decision processes', *Journal of Artificial Intelligence Research*, 22, pp. 423–455. doi: 10.1613/jair.1497.

[20] Campbell, T., Johnson, L. and How, J. P. (2013) 'Multiagent allocation of Markov decision process tasks', in *Proceedings of the American Control Conference*. IEEE, pp. 2356–2361. doi: 10.1109/acc.2013.6580186.

[21] Nemhauser, G. L., Wolsey, L. A. and Fisher, M. L. (1978) 'An analysis of approximations for maximizing submodular set functions-I', *Mathematical Programming*, 14(1), pp. 265–294. doi: 10.1007/BF01588971.

[22] Zhao, X., Zong, Q., Tian, B., Zhang, B. and You, M. (2019) 'Fast task allocation for heterogeneous unmanned aerial vehicles through reinforcement learning', *Aerospace Science and Technology*. Elsevier Masson SAS, 92, pp. 588–594. doi: 10.1016/j.ast.2019.06.024.

[23] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. and Riedmiller, M. (2013) 'Playing Atari with Deep Reinforcement Learning', pp. 1–9. Available at: http://arxiv.org/abs/1312.5602.

[24] Krause, A. and Golovin, D. (2011) 'Submodular function maximization', *Tractability*, 9781107025, pp. 71–104. doi: 10.1017/CBO9781139177801.004.