Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Threshold Bundle-based Task Allocation for Multiple Aerial Robots

Teng Li * **Hyo-Sang Shin** * **Antonios Tsourdos** *

* *School of Aerospace, Transport and Manufacturing, Cranfield University, Cranfield MK43 0AL, UK (e-mail: tengli@cranfield.ac.uk, h.shin@cranfield.ac.uk, and a.tsourdos@cranfield.ac.uk).*

Abstract: This paper focuses on the large-scale task allocation problem for multiple Unmanned Aerial Vehicles (UAVs). One of the great challenges with task allocation is the NP-hardness for both computation and communication. This paper proposes an efficient decentralised task allocation algorithm for multiple UAVs to handle the NP-hardness while providing an optimality bound of solution quality. The proposed algorithm can reduce computational and communicating complexity by introducing a decreasing threshold and building task bundles based on the sequential greedy algorithm. The performance of the proposed algorithm is examined through Monte-Carlo simulations of a multi-target surveillance mission. Simulation results demonstrate that the proposed algorithm achieves similar solution quality compared with benchmark task allocation algorithms but consumes much less running time and consensus steps.

Keywords: Multiple UAVs, task allocation, submodular maximisation, threshold bundle, multi-target surveillance.

1. INTRODUCTION

Multiple Unmanned Aerial Vehicles (UAVs) have been frequently applied in various applications because of their obvious advantages over single UAV, such as stronger adaptability, increased area coverage, and improved reliability. Typical applications include large-area surveillance [Capitan et al. (2016); Li and Duan (2017); Gu et al. (2018)], precision agriculture [Milics (2019)], and search and rescue [Kurdi et al. (2016)].

Cooperation among UAVs is key to their mission success. Finding an effective task execution scheme promptly, which is termed as task allocation, is the foundation of successful cooperation. The task allocation problem considered in this paper is maximising an overall objective function that is the sum of all individual objective function values while satisfying the constraint that one task can only be allocated to one UAV.

The main issue with task allocation is the NP-hardness [Shin and Segui-Gasco (2016)], which means that finding the optimal solution requires exponential time. It is intractable to calculate the optimal solution in large-scale applications. Therefore, developing efficient decentralised task allocation algorithms that can provide a suboptimal solution with low computational and communicating complexity is the leading research direction.

Various approaches have been developed to handle the NP-hardness of the task allocation problem, such as the heuristic approach [Bai et al. (2018); Boveiri (2016); Otte et al. (2019)] and the approximation approach [Qu et al. (2015); Ding and Castanón (2017); Seo et al. (2018)]. Heuristic algorithms can deliver feasible task allocation solutions within certain convergence steps. However, this kind of algorithms cannot provide any optimality bound of the solution quality. Approximation algorithms can provide optimality bounds of solution quality when the objective functions meet the condition of submodularity [Feldman et al. (2017)]. Submodularity is a ubiquitous feature in the real-world combinatorial optimisation problems. The Sequential Greedy Algorithm (SGA) provides an optimality guarantee of at least 1/2 of the optimal solution for maximising monotone submodular objective functions subject to partition matroid constraints [Nemhauser et al. (1978)]. This paper focuses on the algorithms that leverage the approximation approach.

Effective cooperation of UAVs requires a proper network architecture. In a centralised network, all UAVs need to communicate with the ground station or a UAV leader. This kind of networks highly rely on a single entity and a failure to this entity could cause the failure of the entire team. A decentralised network can enhance the reliability and increase the range of coverage of the UAV team. Therefore, the network architecture adopted in this paper is a decentralised one.

Extensive works have utilised the decentralised approximation approach to handle the NP-hardness of computation. [Qu et al. (2015)] solved the decentralised task allocation problem for a large group of satellites using distributed SGA. [Williams et al. (2017)] considered the intersection of multiple matroid constraints in the task allocation of a surveillance mission. [Sun et al. (2019)] solved the multi-agent coverage problem using distributed SGA with the consideration of obstacles and the curvature of submodularity. The approximation approach is also applied in many other areas, such as search and localisation [Ding and Castanón (2017)], and sensor networks [Kumar et al. (2017); Corah and Michael (2018)].

However, a new challenge arises in the decentralised architecture, which is the communicating complexity. UAVs need to communicate their local information with each other and make a consensus. Too many consensus steps could cause inefficiency of the task allocation process.

Several works have considered handling the NP-hardness of communication using the approximation approach. The Consensus-Based Bundle Algorithm (CBBA) developed in [Choi et al. (2009)] is one of the most widely accepted decentralised task allocation algorithms. CBBA can reduce consensus steps hence relaxing the communicating complexity by building task bundles. However, if any conflict appears, UAVs need to rebuild their bundles continually, which incurs increased computational complexity. [Qu et al. (2019)] came up with an assumption of admissible task sets. In this case, a large group of earth-observing satellites could achieve an undiminished optimality bound of the task allocation solution quality even though only local information and communication were available. The assumption is reasonable for homogeneous and static agents but intractable to be applied directly for general heterogeneous moving UAVs. Our previous work [Li et al. (2019b)] proposed an efficient decentralised task allocation algorithm DTTA that enabled parallel allocation by leveraging a decreasing threshold [Badanidiyuru and Vondrák (2014); Buchbinder et al. (2016)]. DTTA could reduce both computational and communicating complexity significantly compared with decentralised SGA and CBBA.

This paper extends DTTA by building task bundles based on the decreasing threshold [Badanidiyuru and Vondrák (2014); Buchbinder et al. (2016)]. The proposed algorithm, Threshold Bundle-based Task Allocation (TBTA), can further reduce consensus steps hence further releasing communicating complexity compared with DTTA. In each iteration of DTTA, each UAV select one qualified task whose marginal value is no less than the current threshold based on its local information then stops searching. While in each iteration of TBTA, each UAV continues searching after it finds one qualified task and adds this task to its bundle until it cannot find any other qualified tasks. Then, it coordinates with other UAVs with its task bundle instead of one task. Therefore, during one consensus step, more tasks could be allocated with TBTA than DTTA.

Moreover, in CBBA, the bundle will become invalid if the first task of the bundle is removed because of the conflict with other UAVs' selections. Then, UAVs need to rebuild their bundles with consideration of the tasks released from the invalid bundles of previous iterations. On the contrary, the bundle in TBTA is still valid even if the first task of the bundle is removed. This is because, if a task is removed from the bundle, the marginal values of the following tasks in this bundle will not decrease according to the feature of submodularity. Thus, all tasks in the bundles are allocated to UAVs during this iteration. No task will remain to the next iteration for reevaluation. Therefore, TBTA is more efficient than CBBA.

The performance of the proposed algorithm TBTA is compared with benchmark task allocation algorithms through Monte-Carlo simulations of a multi-target surveillance mission adopted from [Li et al. (2019a)]. Simulation results indicate that TBTA achieves almost the same solution quality with benchmark algorithms but spends the fewest consensus steps. Moreover, TBTA and DTTA consume much less running time and consensus steps than SGA and CBBA. TBTA consumes fewer consensus steps but slightly more running time than DTTA.

2. PRELIMINARIES

This section provides relevant definitions and concepts that are related to the proposed task allocation algorithm. These concepts can also be found in our previous work [Li et al. (2019a)].

Definition 1. (Task Allocation). Task allocation refers to allocating a group of tasks $\mathcal{T}$ to a group of UAVs $\mathcal{A}$ to maximise the total value measured as

$$f(\mathcal{T}, \mathcal{A}) = \sum_{a=1}^{|\mathcal{A}|} f_a(\mathcal{T}_a), \tag{1}$$

where f_a is the objective function for UAV a, $\mathcal{T}_a$ is a set that contains all the tasks allocated to UAV a. The constraint is that one task can only be allocated to one UAV but each UAV can take multiple tasks.

Definition 2. (Submodularity [Feldman et al. (2017)]) $\mathcal{N}$ is a finite set. A real-valued set function $f : 2^{\mathcal{N}} \to \mathbb{R}$ is *submodular* if, for all $X, Y \subseteq \mathcal{N}$,

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

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

$$f(A \cup \{u\}) - f(A) \geq f(B \cup \{u\}) - f(B). \tag{2}$$

Eqn. (2) is also known as the diminishing return which is an important property of submodular functions. Specifically, the marginal value of a given element u will never increase as more elements have already been selected.

Definition 3. (Marginal value [Feldman et al. (2017)]) For a set function $f : 2^{\mathcal{N}} \to \mathbb{R}$, $S \subseteq \mathcal{N}$ and $u \in \mathcal{N}$, define the *marginal value* of f at S with respect to u as

$$\Delta f(u|S) := f(S \cup \{u\}) - f(S).$$

Definition 4. (Monotonicity [Feldman et al. (2017)]) A set function $f : 2^{\mathcal{N}} \to \mathbb{R}$ is *monotone* if, for every $A \subseteq B \subseteq \mathcal{N}$, $f(A) \leq f(B)$. And f is *non-monotone* if it is not monotone.

The objective functions considered in this paper are normalised (i.e. $f(\emptyset) = 0$), non-negative (i.e. $f(S) \geq 0$ for all $S \subseteq \mathcal{N}$), monotone, and submodular.

Definition 5. (Matroid [Badanidiyuru and Vondrák (2014)]) A matroid is a pair $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ where $\mathcal{N}$ is a finite set and $\mathcal{I} \subseteq 2^{\mathcal{N}}$ is a collection of independent sets, satisfying:

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

Specifically, the matroid constraints consist of uniform matroid constraint and partition matroid constraint. The uniform matroid constraint is also named as cardinality constraint which is a special case of matroid constraint. In cardinality constraints, any subset $S \subseteq \mathcal{N}$ satisfying $|S| \leq k$ is independent. Partition matroid constraint means that a subset S can contain at most a certain number of elements from each partition.

According to Definition 1, at most one task-UAV pair from the task-UAV pairs that are related to this specific task can be selected. If all task-UAV pairs are considered as a ground set (i.e. $\mathcal{N} := \mathcal{T} \times \mathcal{A}$) and each task-UAV pair as an element of the ground set (i.e. $u_{j,a} := j \times a \; \forall j \in \mathcal{T}, a \in \mathcal{A}$), the task allocation problem in this paper can be handled as submodular maximisation subject to a partition matroid constraint.

3. ALGORITHM AND ANALYSIS

This section describes and analyses the proposed decentralised task allocation algorithm TBTA.

3.1 Algorithms

The proposed algorithm TBTA, which is presented in Algorithm 3, mainly has three phases, i.e., initialisation phase, bundle construction phase, and bundle coordination phase. The bundle construction phase and the bundle coordination phase are described as subfunctions in Algorithm 1 and Algorithm 2, respectively.

Algorithm 1 *BuildBundle* for UAV a

Input: $f_a : 2^{\mathcal{T}} \rightarrow \mathbb{R}_{\geq 0}, \mathcal{N}_a, \mathcal{T}_a, \theta$
Output: $\mathcal{B}_a$

1: $\mathcal{B}_a \leftarrow \emptyset$
2: **for** $j \in \mathcal{N}_a$ **do**
3:     **if** $\Delta f_a(j|\mathcal{T}_a \oplus \mathcal{B}_a) \geq \theta$ **then**
4:         $\mathcal{B}_a \leftarrow \mathcal{B}_a \oplus j$
5:     **end if**
6: **end for**
7: **return** $\mathcal{B}_a$

Algorithm 2 *BundleCoor* for UAV a

Input: $a, \mathcal{T}_a, \mathcal{B}_a$
Output: $\mathcal{T}_a, J$

1: $J \leftarrow \emptyset$
2: Send $\mathcal{B}_a$ to all UAVs $i \in \mathcal{A}$.
3: Receive $\mathcal{B}_i$ from all UAVs $i \in \mathcal{A}$.
4: **while** $\exists \mathcal{B}_i \neq \emptyset$ **do**
5:     **for** $i \in \mathcal{A}$ **do**
6:         **if** $\mathcal{B}_i \neq \emptyset$ **then**
7:             **if** $\mathcal{B}_i[0] \notin J$ **then**
8:                 $\mathcal{T}_i \leftarrow \mathcal{T}_i \oplus \mathcal{B}_i[0]$
9:                 $J \leftarrow J \oplus \mathcal{B}_i[0]$
10:             **end if**
11:             $\mathcal{B}_i \leftarrow \mathcal{B}_i \backslash \{\mathcal{B}_i[0]\}$
12:         **end if**
13:     **end for**
14: **end while**
15: **return** $\mathcal{T}_a, J$
// $\mathcal{B}_i[0]$ is the first element of the bundle $\mathcal{B}_i$

In the initialisation phase (Algorithm 3, lines 1 $\sim$ 4), UAV a conducts preparatory works. UAV a searches the task ground set and finds the locally largest marginal value ω_a^{max}. Then it makes a consensus with all other UAVs through the $MaxCons$ function to find the globally largest marginal value $\omega_{a^*}^{max}$ (stored as d) and set it as the initial value of the threshold θ. The threshold value is decreasing overtime. The terminal value of the threshold is set as $\frac{\epsilon}{r}d$, where $\epsilon \in (0, 1)$ is the threshold decreasing parameter, r is the number of tasks.

Algorithm 3 decentralised TBTA for UAV a

Input: $f_a : 2^{\mathcal{T}} \rightarrow \mathbb{R}_{\geq 0}, \mathcal{T}, \mathcal{A}, \epsilon$
Output: A set $\mathcal{T}_a \subseteq \mathcal{T}$

1: $\mathcal{T}_a \leftarrow \emptyset$, $\mathcal{N}_a \leftarrow \mathcal{T}$
2: $\omega_a^{max} \leftarrow \max_{j_a \in \mathcal{N}_a} \Delta f_a(j_a|\mathcal{T}_a)$
3: $a^*, \omega_{a^*}^{max} \leftarrow MaxCons(a, \omega_a^{max})$
4: $d \leftarrow \omega_{a^*}^{max}, \theta \leftarrow d, r \leftarrow |\mathcal{T}|$
5: **while** $\theta \geq \frac{\epsilon}{r} d$ **do**
6:     $\mathcal{B}_a \leftarrow BuildBundle(f_a, \mathcal{N}_a, \mathcal{T}_a, \theta)$
7:     $\mathcal{T}_a, J \leftarrow BundleCoor(a, \mathcal{T}_a, \mathcal{B}_a)$
8:     **if** $J \neq \emptyset$ **then**
9:         $\mathcal{N}_a \leftarrow \mathcal{N}_a \backslash J$
10:     **else**
11:         $\theta \leftarrow (1 - \epsilon)\theta$
12:     **end if**
13: **end while**
14: **return** $\mathcal{T}_a$

The function $MaxCons$ in Algorithm 3 is the maximum consensus function. UAV a sends the locally largest marginal value ω_a^* together with the corresponding UAV id a to all other UAVs. Meanwhile, this UAV also receives this kind of information from all other UAVs. After UAV a has gathered the information from all other UAVs, it will find the globally largest marginal value $\omega_{a^*}^{max}$. It is proved that $MaxCons$ can reach a consensus within finite time [Giannini et al. (2016)]. Note that, each UAV uses this function only for one time.

In the bundle construction phase, UAV a builds its bundle given the current threshold θ. The notation $\oplus$ represents the operation of appending the right element to the end of the left element. UAV a evaluates all the remaining tasks in $\mathcal{N}_a$ and appends the qualified tasks to its task bundle $\mathcal{B}_a$ one by one. A *qualified task* means that the marginal value of this task is no less than the current threshold θ. Note that the marginal value is calculated given the combination of $\mathcal{T}_a$ and the updated $\mathcal{B}_a$.

In the bundle coordination phase, UAVs coordinate with each other using their bundles and allocate tasks. The notation J is an auxiliary set that contains all the tasks that have already been allocated to UAVs. UAVs check the first tasks from all non-empty bundles. If the task $\mathcal{B}_i[0]$ is not in J, then append it to the corresponding $\mathcal{T}_a$ and J, respectively (Algorithm 2 lines 7 $\sim$ 10). Here, $\mathcal{B}_i[0]$ is the first element of the bundle $\mathcal{B}_i$ for UAV a. Then, remove this task from the bundle $\mathcal{B}_i$ (Algorithm 2 line 11). All tasks that appear in this phase have marginal values no less than the current threshold. At the end of this phase, all of them will be allocated, which enables parallel allocation. After the bundle coordination, UAV a removes the tasks appearing in J from its remaining task set $\mathcal{N}_a$ if J is not empty (Algorithm 3 lines 8 $\sim$ 9). $J = \emptyset$ means that no UAV can build a bundle under the current threshold, which triggers the decrease of the threshold (Algorithm 3 lines 10 $\sim$ 11).

3.2 Analysis

TBTA requires a fewer number of function evaluations, thus consumes less running time, Compared with CBBA. There are two main reasons.

On the one hand, TBTA builds bundles differently with CBBA. In CBBA, each UAV needs to evaluate all the remaining tasks to find the best task that can provide the largest marginal value. After adding this task to its bundle, this UAV evaluates all the remaining tasks again to find the next best task given its current selection and task bundle. This means that UAVs with CBBA need to reevaluate all the remaining tasks every time before they add a new task into their bundles. However, in TBTA, UAVs evaluate all the remaining tasks only for one round to build their bundles. Once a UAV finds a qualified task, it will add this task to its bundle directly.

On the other hand, the bundles in CBBA could become invalid during each consensus step, but the bundles in TBTA are always valid. As mentioned in [Choi et al. (2009)], the bundles are built based on the largest marginal values. If one task is removed from the bundle because of the conflict with other UAVs, the following tasks in this bundle will become invalid. Then, this UAV needs to rebuild its bundle. However, in TBTA, the marginal values of the following tasks are still no less than the current threshold because of submodularity. Therefore, TBTA requires less number of objective function evaluations than CBBA.

TBTA has the same theoretical performance guarantee with DTTA in terms of both solution quality and computational complexity. In both DTTA and TBTA, UAVs evaluate all the remaining tasks only for one round to find all qualified tasks given each threshold. Therefore, TBTA and DTTA achieve the same theoretical optimality guarantee and consume the same number of objective function evaluations in the worst case.

The theoretical performance of TBTA is summarised in Theorem 1.

Theorem 1. The proposed algorithm TBTA achieves an optimality bound of at least $1/2 - \epsilon$ for maximising monotone submodular objective functions with computational complexity of $O(\frac{r}{\epsilon} \ln \frac{r}{\epsilon})$ for each UAV, where $r = |\mathcal{T}|$ is the number of tasks, ϵ is the threshold decreasing parameter.

Since TBTA has the same theoretical performance with DTTA, please refer to our previous work [Li et al. (2019b)] for the proof of Theorem 1.

4. NUMERICAL SIMULATIONS

This section validates the proposed task allocation algorithm TBTA via numerical simulations. The simulation scenario is adopted from [Segui-Gasco et al. (2015)] in the monotone case, which is a multi-target surveillance mission using multiple heterogeneous UAVs.

In the simulation scenario, it is assumed that a group of UAVs $a \in \mathcal{A}$ equipped with different sensors automatically carry out the surveillance mission of a set of heterogeneous task points that are randomly located on a $L \times L$ 2-D space. Here, different sensors can represent different cameras with different technical specifics.

The surveillance mission aims to maximise the reward that is measured by an objective function while minimising the calculation time and communication burden. The constraint is that one task can only be allocated to one UAV.

The objective function of the surveillance mission is a coverage-type function. Different tasks are marked with an *task importance factor* v_j according to their values. Different sensors are suitable for different tasks. The *task-UAV fitness factor* m_{aj} represents the match fitness between the abilities of UAV a and the requirements of task j. The utility of the tasks $j \in \mathcal{T}_a$ is measured as the sum of the product of m_{aj} and v_j. For the tasks $j \notin \mathcal{T}_a$, we add an exponentially decaying term related to the shortest distance between task j and tasks in $\mathcal{T}_a$ which is denoted as $d_{\min}(j, \mathcal{T}_a)$. When UAV a is carrying out a task at the location of this task, it can partially serve another one nearby. The objective function for UAV a is

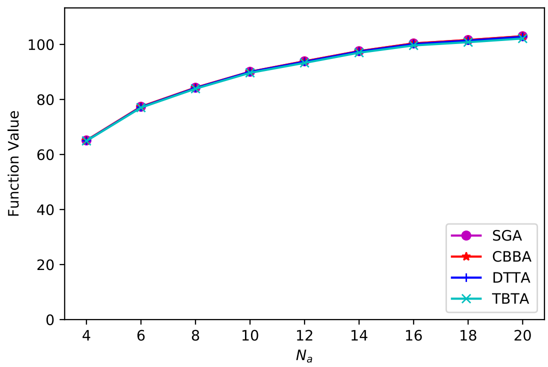
$$f_a(\mathcal{T}_a) = \sum_{j \in \mathcal{T}_a} m_{aj} v_j + \sum_{j \notin \mathcal{T}_a} m_{aj} v_j e^{-\frac{d_{\min}(j, \mathcal{T}_a)}{d_0}}, \quad (3)$$

where d_0 is a reference distance. The overall objective function of the surveillance mission is obtained by combining Eqns. (3) and (1):

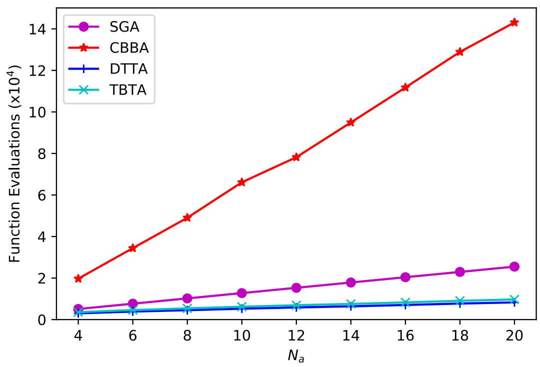
$$f(\mathcal{T}, \mathcal{A}) = \sum_{a \in \mathcal{A}} \left[\sum_{j \in \mathcal{T}_a} m_{aj} v_j + \sum_{j \notin \mathcal{T}_a} m_{aj} v_j e^{-\frac{d_{\min}(j, \mathcal{T}_a)}{d_0}} \right].$$

We conduct Monte-Carlo simulations to validate the proposed algorithm and compare its performance with benchmark decentralised task allocation algorithms. In the simulations, 50 tasks and varying numbers of UAVs are randomly located on a $L \times L$ 2-D space, where $L = 10km$. The numbers of UAVs are increased from 4 to 20, denoted by N_a. The importance factors of tasks are chosen from uniformly random numbers $v_j \in [0.6, 1.0]$. Each task-UAV fitness factor is set as a uniformly random number $m_{aj} \in [0.5, 1.0]$. Set the threshold decreasing parameter $\epsilon = 0.1$, the reference distance as $d_0 = 1km$. The running time is measured as the number of objective function evaluations, which is independent on the computer status. We run 100 rounds of simulations and get the average values. The performances of SGA, CBBA, DTTA, and TBTA are compared with SGA as a baseline.

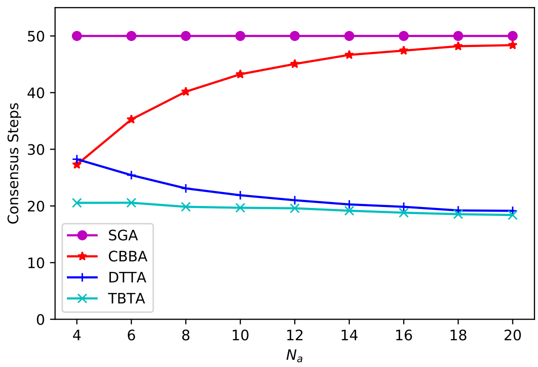
The Monte-Carlo simulation results are demonstrated in Fig. 1. According to Fig. 1 (a), it is clear that all these four algorithms achieve almost the same solution quality. Overall, TBTA performs the best in terms of reducing communicating complexity. CBBA can effectively reduce consensus steps, but it consumes obviously increased running time compared with SGA. DTTA and TBTA can significantly reduce both running time and the number of consensus steps. Fig. 1 (f) shows that, when there are 20 UAVs, TBTA only consume 36.8% of consensus steps and 38% of running time compared with SGA. Note that, as the number of UAVs goes up, the percentages of function values achieved by DTTA and TBTA almost stay stable, but the percentages of running time and consensus steps are decreasing with SGA as a baseline. This means that



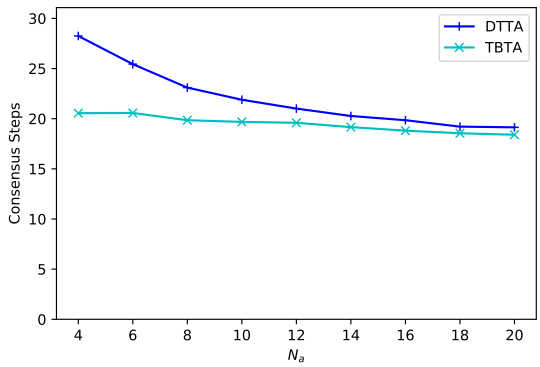
(a) Function values



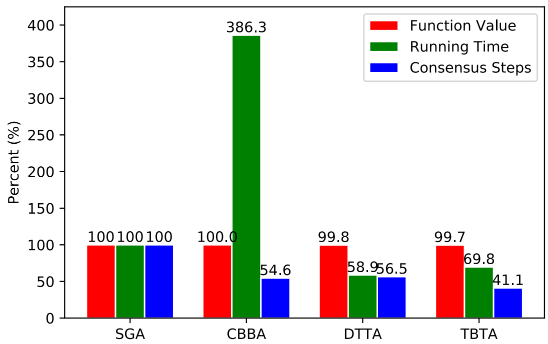
(b) Running time



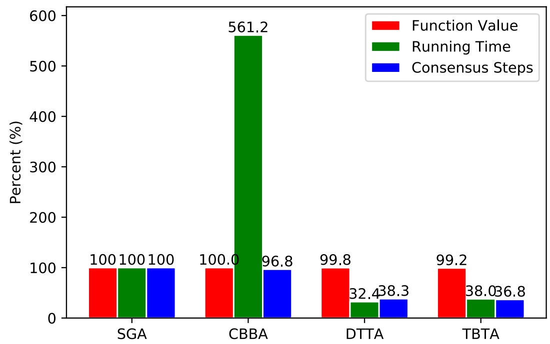
(c) Consensus steps



(d) Consensus steps



(e) Ratio comparison $N_a = 4$



(f) Ratio comparison $N_a = 20$

Fig. 1. Performance comparison of different task allocation algorithms (N_a is the number of agents)

the performances of DTTA and TBTA are getting better as the scale of the task allocation problem increases.

Comparing TBTA with DTTA, it is clear that TBTA can further reduce the number of consensus steps based on DTTA. Fig. 1 (d), (e), and (f) indicate that TBTA consumes fewer numbers of consensus steps than DTTA, although it spends a bit more running time. As the number of UAVs increases, the differences in the performances of DTTA and TBTA become smaller. At a high ratio of the number of tasks versus the number of UAVs, TBTA obviously outperforms other algorithms in terms of releasing communicating burden. Additionally, the numbers of consensus steps consumed by DTTA and TBTA decrease as the number of UAVs increases. The reason is that, more tasks can be allocated during each consensus step as more UAVs participating. This is a beneficial feature in the large-scale decentralise task allocation problems where reducing communicating complexity is one of the significant challenges.

5. CONCLUSION

This paper proposed a decentralised algorithm, which is named as Threshold Bundle-based Task Allocation (TBTA), for the large-scale task allocation problems. The performance of the proposed algorithm was analysed theoretically and validated through Monte-Carlo simulations.

TBTA could not only relax the computational complexity but also reduce the communicating complexity for the decentralised UAV systems compared with benchmark algorithms. Simulation results revealed that TBTA consumed the fewest number of consensus steps.

One future work could be applying the lazy strategy that was developed in our previous work [Li et al. (2019b)] to further reduce the running time and the number of consensus steps. Another future work could be validating the proposed algorithm with physical experiments using multiple drones.

REFERENCES

Badanidiyuru, A. and Vondrák, J. (2014). Fast algorithms for maximizing submodular functions. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 1497–1514. SIAM.

Bai, X., Yan, W., Ge, S.S., and Cao, M. (2018). An integrated multi-population genetic algorithm for multi-vehicle task assignment in a drift field. *Information Sciences*, 453, 227–238.

Boveiri, H.R. (2016). A novel aco-based static task scheduling approach for multiprocessor environments. *International Journal of Computational Intelligence Systems*, 9(5), 800–811.

Buchbinder, N., Feldman, M., and Schwartz, R. (2016). Comparing apples and oranges: Query trade-off in submodular maximization. *Mathematics of Operations Research*, 42(2), 308–329.

Capitan, J., Merino, L., and Ollero, A. (2016). Cooperative decision-making under uncertainties for multi-target surveillance with multiples uavs. *Journal of Intelligent & Robotic Systems*, 84(1-4), 371–386.

Choi, H.L., Brunet, L., and How, J.P. (2009). Consensus-based decentralized auctions for robust task allocation. *IEEE Transactions on Robotics*, 25(4), 912–926.

Corah, M. and Michael, N. (2018). Distributed submodular maximization on partition matroids for planning on large sensor networks. In *2018 IEEE Conference on Decision and Control (CDC)*, 6792–6799. IEEE.

Ding, H. and Castanón, D. (2017). Multi-agent discrete search with limited visibility. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 108–113. IEEE.

Feldman, M., Harshaw, C., and Karbasi, A. (2017). Greed is good: Near-optimal submodular maximization via greedy optimization. In *Proceedings of the 2017 Conference on Learning Theory (COLT)*, volume 65, 1–27. PMLR.

Giannini, S., Petitti, A., Di Paola, D., and Rizzo, A. (2016). Asynchronous max-consensus protocol with time delays: Convergence results and applications. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 63(2), 256–264.

Gu, J., Su, T., Wang, Q., Du, X., and Guizani, M. (2018). Multiple moving targets surveillance based on a cooperative network for multi-uav. *IEEE Communications Magazine*, 56(4), 82–89.

Kumar, R.R., Varakantham, P., and Kumar, A. (2017). Decentralized planning in stochastic environments with submodular rewards. In *AAAI*, 3021–3028.

Kurdi, H., How, J., and Bautista, G. (2016). Bio-inspired algorithm for task allocation in multi-uav search and rescue missions. In *AIAA Guidance, Navigation, and Control Conference*, 1377.

Li, P. and Duan, H. (2017). A potential game approach to multiple uav cooperative search and surveillance. *Aerospace Science and Technology*, 68, 403–415.

Li, T., Shin, H.S., and Tsourdos, A. (2019a). Efficient decentralized task allocation for uav swarms in multi-target surveillance missions. In *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, 61–68. IEEE.

Li, T., Shin, H.S., and Tsourdos, A. (2019b). Threshold greedy based task allocation for multiple robot operations. *arXiv preprint arXiv:1909.01239*.

Milics, G. (2019). Application of uavs in precision agriculture. In *International Climate Protection*, 93–97. Springer.

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), 265–294.

Otte, M., Kuhlman, M.J., and Sofge, D. (2019). Auctions for multi-robot task allocation in communication limited environments. *Autonomous Robots*, 1–38.

Qu, G., Brown, D., and Li, N. (2015). Distributed greedy algorithm for satellite assignment problem with submodular utility function. *IFAC-PapersOnLine*, 48(22), 258–263.

Qu, G., Brown, D., and Li, N. (2019). Distributed greedy algorithm for multi-agent task assignment problem with submodular utility functions. *Automatica*, 105, 206–215.

Segui-Gasco, P., Shin, H.S., Tsourdos, A., and Segui, V. (2015). Decentralised submodular multi-robot task allocation. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2829–2834. IEEE.

Seo, M.G., Shin, H.S., and Tsourdos, A. (2018). Task allocation in agricultural remote sensing applications using submodular maximization algorithm. In *2018 37th Chinese Control Conference (CCC)*, 6860–6865. IEEE.

Shin, H.S. and Segui-Gasco, P. (2016). Uav swarms: Decision-making paradigms. *Unmanned Aircraft Systems*, 397.

Sun, X., Cassandras, C.G., and Meng, X. (2019). Exploiting submodularity to quantify near-optimality in multi-agent coverage problems. *Automatica*, 100, 349–359.

Williams, R.K., Gasparri, A., and Ulivi, G. (2017). Decentralized matroid optimization for topology constraints in multi-robot allocation problems. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 293–300. IEEE.