2023 21st International Conference on Advanced Robotics (ICAR)
5-8 December 2023. Abu Dhabi, UAE

Decentralized Multi-UAV Trajectory Task Allocation in Search and Rescue Applications

Kasper A. R. Grøntved and Ulrik P. S. Lundquist and Anders Lyhne Christensen

***Abstract*— Multi-UAV systems have significant potential to enhance search and rescue (SAR) operations, since a search area can be covered faster than current approaches when multiple UAVs operate in parallel. While recent advancements within the field of multi-robot coverage planning have yielded promising results, current algorithms are predominately centralized. In this paper, we present a generalization of the well-known decentralized consensus-based bundle algorithm (CBBA), that enables efficient task allocation in multi-UAV SAR operations. The generalized algorithm considers tasks as trajectories between two points where the traversal direction for each task is optimized in the task allocation process. We carry out a series of simulation-based experiments on benchmark problems and compare our results to a state-of-the-art centralized solution. We find that our novel decentralized approach yields times to completion similar to those achieved with a centralized coverage path planning approach, with only 1.9% overhead cost. We furthermore find that our approach performs 6% better than point allocations while scaling well with the number of UAVs involved in the search effort.**

I. Introduction

Unmanned aerial vehicles (UAVs) are being employed in an increasingly broad range of domains, including SAR operations [1], [2], due to their high degree of freedom and capabilities of moving mostly unobstructed through an environment [3]. Currently, single-unit systems are primarily used, and they tend to be either remote-controlled or have limited autonomous decision-making authority. Multi-UAV systems, on the other hand, have significant potential as the UAVs can operate in parallel and thus cover a search area faster than a single-unit system. However, for efficient multi-UAV operation, a system needs to be controllable by a single operator, which means that the system must have a relatively high level of autonomy, such as the ability to automatically allocate search trajectories to UAVs given a high-level mission specification [4].

In SAR operations, the search area is commonly specified as a polygon, see Fig. 1 for an example. As can be seen in the figure, the search area is further subdivided into a number of regions, and a set of trajectories (*trajectory tasks*, shown in red) is defined for each sub-region. In the example, there are a total of 43 trajectory tasks. The task allocation problem then consists of allocating a set of UAVs to carry out all trajectory tasks, such that the time to completion is minimized [5], [6]. This problem is also known as *coverage path planning* [7]. For single-unit systems, this often results in the lines being covered in a boustrophedon pattern [7], [8].

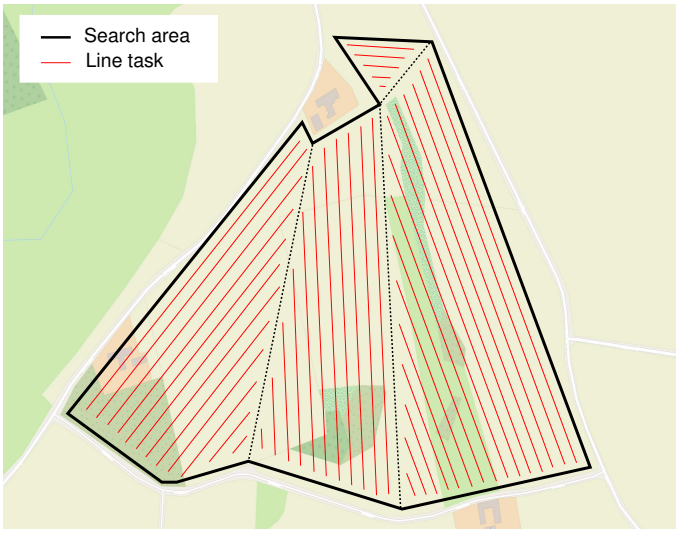


Fig. 1: Example of a search area for a SAR operation given as a polygon (thick black line). The search area is divided into four sub-regions (dashed lines), and trajectory tasks (in red) are defined for each sub-region.

The problem of coverage planning is a widely studied field in robotics, see [8]–[11] for examples. Nevertheless, most approaches are centralized and thus require reliable, continuous connectivity between a base station and all UAVs, which may not be available in SAR operations that are conducted in large search areas and in remote locations where communication infrastructure is sparse or non-existent. Also, in multi-robot coverage planning, it is a common assumption that the area and the available units remain static throughout a mission. However, since SAR operations often are conducted in a dynamic and unstructured environment, the search area and sub-regions often change during an operation [12].

In this paper, we focus on a decentralized algorithm called the *consensus-based bundle algorithm* (CBBA) [13]. CBBA is an auction-based algorithm, which iterates between a bundle-construction phase, in which tasks are assembled into bundles, and a conflict-resolution phase, in which consensus is reached. The CBBA does not require direct connectivity between all UAVs in a system to converge to a conflict-free task allocation. In the original formulation of CBBA, tasks are modelled as points, and the algorithm does thus not take the trajectorial nature of SAR tasks (lines in the simplest case) into account.

This work is supported by the Innovation Fund Denmark for the project DIREC (9142-00001B).

All authors are with the SDU UAS Center, MMMI, University of Southern Denmark. E-mail: {kang,ups,andc}@mmmi.sdu.dk

2023 21st International Conference on Advanced Robotics (ICAR) | 979-8-3503-4229-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICAR58858.2023.10406912

The contribution of this paper is a generalization of CBBA [13] to handle trajectory tasks to enable decentralized coverage path planning in SAR scenarios. The travel direction for each trajectory task is considered in the task evaluation, which is used when the UAVs bid for tasks. We establish the efficacy of our algorithm on a coverage path planning benchmark dataset by converting it into trajectory task allocation problems. Experiments are carried out to show that the enhancement can improve the baseline CBBA algorithm by an average of 6% in a range of multi-UAV system sizes and environments. Comparative experiments show that our decentralized planner has low overhead (only 1.9% overhead cost compared to the state-of-the-art centralized heuristic solvers for line coverage problems [11]).

II. Related Work

Area coverage algorithms can be categorized based on the assumptions they make on the geometry of an area [1]. One of the more general assumptions is that a search area can be decomposed into convex cells, which can be covered by a search pattern, such as the boustrophedon pattern [7]. This search pattern is widely used for pre-operation planning in SAR operations [8], [11], [14]. Other methods assume that a search area can be covered by a grid, where each square has to be visited to be considered covered. The latter has the downside of complexity, where the number of squares increases quadratically with the size of the search area. A SAR operational area is commonly specified as one or more polygons [15], along with any interior regions (e.g. obstacles) that robots should not enter. The area is divided into non-intersecting convex polygons called *cells* using a decomposition technique [3], [7], [8]. This approach is especially popular since it is easily understandable by human operators.

The mTSP (Multiple Travelling Salesman Problem) is a well-known NP-hard combinatorial optimization problem [16], for which several heuristic-based methods have been proposed for solving task allocation problems for multi-robot systems, see for instance [8], [11], [17]. In a recent study [18], the authors proposed a novel environment partitioning algorithm and modelled it as a mTSP to solve the multi-robot coverage problem. By minimizing turns, the authors showed that coverage time was reduced significantly for teams of 1-5 robots in 25 indoor environments. In [10], the MEM heuristic algorithm was proposed to solve large-scale multi-robot line coverage problems in a centralized manner. In [11], MEM was then combined with an approach that considers asymmetric costs for *servicing* (covering a task) and *deadheading* (travelling to a task) to model environmental features such as wind. This allows for solving the area coverage problem for capacity-constrained robots using constrained optimisation methods, though still in a centralized manner.

While the above-mentioned algorithms yield promising results, they are centralized. For multi-UAV SAR in real-world conditions, such a centralized approach would require the planning to be done by a single centralized entity, that can reliably and frequently communicate with all the UAVs involved in the search effort. For large-scale SAR operations which often take place in areas where communication infrastructure may be damaged or absent, such communication requirements may be challenging to meet. Also, the reliance on a central entity constitutes a single point of failure, which is a major drawback in SAR operations where robustness is particularly important.

A decentralized system can potentially overcome the shortcomings associated with a centralized approach. By relying on peer-to-peer communication [19], the task allocation algorithm can run on each agent simultaneously and the allocation is obtained through the exchange of information between the agents [20], [21]. The single point of failure can be eliminated by using decentralized architectures such as auction-based methods [13]. Some of the simplest auction-based methods are single-assignment approaches where each agent can only be assigned a single task and where tasks are auctioned one-by-one [12].

One can achieve a near-optimal allocation of a set of tasks when using *single-round combinatorial auctions* [22], where bids are calculated based on sets of tasks (*bundles*) instead of individual tasks. Each agent's bid corresponds to the minimum path cost of a bundle, based on the position of the agent. Each agent has to try all permutations of tasks as a bundle, which is computationally expensive since the number of bids grows exponentially with the number of tasks. The above-mentioned methods rely on a central broker or auctioneer, whereas other methods, such as the *consensus-based bundle algorithm* (CBBA) discussed in further detail below, are fully decentralized [13]. CBBA iterates between two phases, (i) *the bundle-construction phase*, where each agent builds its own bundle of tasks; and (ii) *the conflict-resolution phase*, where the agents communicate with their neighbours to reach consensus on who has won which auctions. These phases are repeated until the algorithm converges to a conflict-free assignment of all tasks. Several extensions to CBBA have been proposed to expand its applicability, such as asynchronicity [23], heterogeneous agents [24], time-critical tasks [25] and maximisation of the total number of tasks allocated, thus ensuring the tasks are distributed more evenly [26]. Decentralized methods, such as CBBA, can have a large impact on the execution of a SAR operation because decentralized algorithms have the general advantage of being more robust to changes in the environment and not having to rely on stable communication with a ground station [19].

Current auction-based methods often assume that tasks are independent of one another and that each task can be solved in a single, unique way. This assumption is valid in previous applications of auction-based methods to non-coverage-based SAR problems, such as in [27], where each task corresponds to a site with victims. Each task is thus only a point. However, to solve the coverage path planning problem in SAR for *victim localisation*, the spatial nature of coverage tasks, namely that a path between two points must be travelled to satisfy a trajectory task, and should be taken into account to achieve optimal plans. Given that a

line segment, for instance, can be travelled in two directions, the distance between two tasks depends on the direction in which each of them is covered. In this paper, we therefore propose a generalization of the highly popular decentralized market-based task allocation algorithm, CBBA, where different possible travel directions are explicitly considered in the computation of plan costs. The resulting algorithm enables effective decentralized *coverage planning* for multi-UAV systems in SAR and seamless integration of special costs for generating task bids, such as including the cost of turning in the bid.

III. The Consensus-based Bundle Algorithm

CBBA [13] is a fully decentralized multi-assignment task allocation algorithm, which implements a greedy auction strategy, and enables agents to auction tasks based on task bundles built sequentially. CBBA is a well-known and tested algorithm, that provides solutions with a minimum performance guarantee of 50% optimality, assuming that agents have accurate knowledge about auction winners [13]. The goal of the task allocation problem is to find a conflict-free allocation of tasks to agents which optimizes a scoring function, given a list of N_t tasks and N_u agents, where each agent is able to get assigned L_t tasks [13]. The scoring function assigns a score to a task based on either the time or distance to the task.

CBBA consists of two phases, a bundle-construction phase where each agent, i, greedily generates a bundle of ordered tasks and a conflict-resolution phase where conflicting assignments are identified and resolved through local communication. These two phases alternate until a conflict-free task allocation is obtained and satisfies $N_{\min} \triangleq \min\{N_t, N_u L_t\}$, which means convergence is achieved when either the total capacity, $N_u L_t$, is reached or all the tasks, N_t, have been assigned. In CBBA, every agent maintains two different lists of tasks, the bundle $\mathbf{b}_i$ and their path $\mathbf{p}_i$, where the tasks in the bundle are ordered based on when the task was added during the bundle-construction phase. The path $\mathbf{p}_i$ is an ordered list of non-conflicting tasks added after the conflict-resolution phase, where the sequence is based on the marginal score improvement in Eq. (1), such that the path has an optimal sequence.

$$c_{ij}\left[\mathbf{b}_i\right] = \begin{cases} 0, & \text{if } j \in \mathbf{b}_i \\ \max_{n \le |\mathbf{p}_i|} S_i^{\mathbf{p}_i \oplus_n \{j\}} - S_i^{\mathbf{p}_i}, & \text{otherwise,} \end{cases} \quad (1)$$

where $|\cdot|$ denotes the cardinality of the list and the operation $\oplus_n$ inserts the second list after the nth element in the first list. Furthermore, the task j will not provide any reward if the task is already in the task.

The bundle and path are recursively updated as:

$$\mathbf{b}_i = \mathbf{b}_i \oplus_{\text{end}} \{J_i\}, \quad \mathbf{p}_i = \mathbf{p}_i \oplus_{n_{i,J_i}} \{J_i\}, \quad (2)$$

where J_i is the task resulting in the largest increase of score and n_i is the optimal index in agent i's path.

In the bundle-construction phase, the agents' perception of which agent has the highest bid for a task is important, as an agent will only add a task to its bundle if it believes that it has the largest marginal increase in score. Therefore, in the conflict-resolution phase, each agent communicates both its beliefs about the winning bids, the winning agents, and the timestamps of the last information update from each of the other agents, and potential conflicts are then resolved locally using a set of action rules [13]. When an agent is outbid, the tasks that have been added after the outbid task in the bundle will also be removed, as the tasks which the scores were calculated from are no longer valid. Otherwise, the agent could continue to make decisions based on outdated information, which can lead to degraded performance.

CBBA is a polynomial-time algorithm [13] with the worst-case complexity of the bundle-construction phase for a single agent of:

$$\mathcal{O}\left(N_t L_t\right), \quad (3)$$

and has a convergence guarantee within $\max\{N_t, L_t N_u\} D$ iterations, where D is the network diameter, which can be N_u at most.

IV. Trajectory tasks

Consider a multi-UAV system tasked with performing a SAR operation in a bounded area. The goal is to locate any survivors as quickly as possible and thus minimize coverage time. Assume that the search area, given as one or more polygons, has been divided into line segments according to some method, e.g. [18] or [11]. To cover the search area, each line segment must be travelled once by one of the UAVs, and for non-trivial problems, each UAV must be assigned multiple segments, making the problem a multi-assignment task allocation problem. For the trajectory tasks generalization, we consider a task, j_i, as a subset of two points, an *entry point* and a *departure point*:

$$j_i \triangleq \{u_0, u_1\} \quad (4)$$

The trajectory task abstraction is different from simple point tasks, as we assume that after a task is completed, a UAV is not in the same location as where the task started (unless for the special case for non-linear trajectories where $u_0 = u_1$). For tasks where $u_0 \neq u_1$, such as line segment tasks, the scores calculated by the scoring function are different to those of a point task. The set of tasks that the bidder must evaluate to create a bid frequently exhibits significant synergistic potential. The synergy for an agent is considered positive when the combined cost for executing the tasks together is less than the sum of performing them individually [28]. The synergies between trajectory tasks affect the bid more than that of a point task because the agent will end a trajectory task in a different location than where it started the task. This means that not only the entry point of a task contributes to the synergies between tasks, but also the departure point.

When an agent adds a trajectory task to its bundle, it has to evaluate the marginal change in score for adding this task, by calculating the cost of travelling to the task and the cost of travelling after the task. To maximize the

marginal score gain of acquiring the task, the cost must be calculated for both directions in which a trajectory task can be performed. Therefore, the agents have to be able to permute the trajectory tasks by changing the direction of travel based on this optimization.

A. Trajectory task scoring function

CBBA is a versatile algorithm where task-specific constraints or objectives can be taken into account by modelling the scoring function. Though the scoring function has to satisfy a vital property in CBBA called *diminishing marginal gain* (DMG) [13]. DMG states that a tasks score cannot increase in value as another task is added to the path before it and can also be expressed as:

$$c_{ij}\left[\mathbf{b}_i\right] \geq c_{ij}\left[\mathbf{b}_i \oplus_{\text{end}} \mathbf{b}\right] \tag{5}$$

In this paper, we use the time-discounted reward as the scoring function which can be seen in Eq. (6):

$$S_i^{\mathbf{p}_i} = \sum \lambda_j^{\tau_i^j(\mathbf{p}_i)} \bar{c}_j, \tag{6}$$

where $\tau_i^j(\mathbf{p}_i)$ is the estimated time for agent i to reach task j, on the current path $\mathbf{p}_i$, $\bar{c}_j$ is the static reward for completing the task and $\lambda \in (0, 1)$ the discount factor. In search and rescue scenarios, specific tasks can have different priorities, these priorities can be translated into this static reward, to provide a stronger incentive to acquire urgent tasks. In this paper, we keep the reward constant for all tasks but different rewards can be used to prioritise critical tasks. To better approximate SAR operations, where UAVs perform many sharp turns, which results in the UAV having to decelerate to a near stop, the turn cost is modelled as a velocity-ramp function in the scoring function to ensure correct cost estimates. This is done by calculating the time it takes to travel from one point by using a finite acceleration, and can be approximated using a velocity-ramp function, thereby adding an extra cost of travel between tasks by enforcing a complete stop between tasks.

The time it takes to travel the distance d_i^j, the cumulative distance to the task j in agent i's path $\mathbf{p}_i$ using a maximum velocity and acceleration, v_{max} and a_{max}, can be estimated using a velocity-ramp model as follows [8], [11]:

$$\tau_i^j(\mathbf{p}_i) = \begin{cases} \sqrt{\frac{4d_i^j}{a_{\max}}}, & \text{if } d_i^j < d_a \\ \frac{v_{\max}}{a_{\max}} + \frac{d_i^j}{v_{\max}}, & \text{if } d_i^j \geq d_a \end{cases}, \text{ where } d_a = \frac{v_{\max}^2}{a_{\max}} \tag{7}$$

The velocity ramp in equation IV-A is a conditional function where the case $d_i^j < d_a$ describes the time it takes to travel a distance d_i^j if the maximum velocity is not reached, where the latter case $d_i^j > d_a$ describes the time travelling the distance d_a if the maximum velocity is reached within the distance d_a. When a task is given as a trajectory, the function $\tau_i(\mathbf{p}_i)$ calculates the time it takes for an agent to traverse a set of trajectory tasks, from the *departure point* of the previous task to the *entry point* of the next task, for each task in the path $\mathbf{p}_i$. The direction in which the agent traverses the tasks thus greatly impacts the score of the entire path $S_i^{\mathbf{p}_i}$. This means that when the agents construct their bundles and bids on tasks, the direction of approach has to be taken into account for each task in the local path $\mathbf{b}_i$, resulting in a task no longer only containing a reward $\bar{c}_j$ and a point but employ the trajectory task generalization in Eq. 4.

When an agent i builds its bundle and wants to add the task j to its bundle it will swap the u_0 and u_1 if $\tau(\mathbf{p}_{ij}^{u_1}) < \tau(\mathbf{p}_{ij}^{u_0})$ and maximizing the marginal increase of the scoring function (Eq. (8)):

$$c_{ij}\left[\mathbf{b}_i\right] = \begin{cases} 0, & \text{if } j \in \mathbf{b}_i \\ \max_{n \leq |\mathbf{p}_i|} \max_{dir} S_i^{\mathbf{p}_i \oplus_n \{j_{dir}\}} - S_i^{\mathbf{p}_i}, & \text{otherwise,} \end{cases} \tag{8}$$

where j_{dir} is the direction of approach to the trajectory task j and that results in the highest reward based on the previous tasks and directions, effectively performing a greedy optimization. When the first task is added, the agent calculates the score with respect to its starting position.

Once an agent has won an auction for a task j_i, before consensus, the agent will update its own path list and reverse the candidate task j_i if this provides the highest marginal gain, before inserting it into $\mathbf{p}_i$ at the position in the path which yields the highest marginal gain, as per Eq. (2). The information about whether a task has been reversed or not is not communicated to other agents, as this is not important for the allocation of the tasks, similarly, the agent does not communicate the order of the tasks in their paths, but only the information regarding the actual allocation, and thereby does not increase the communication load of the CBBA algorithm.

V. Simulation and experiments

We first assess the performance of our decentralized trajectory task allocation approach against a state-of-the-art centralized method [11], and we then assess scalability on the AC300 benchmark dataset [8].

In the experiments, we use the discounted reward scoring function in Eq. (6) with the discount factor $\lambda = 0.95$. The capacity of each robot L_t is the maximum flight time in seconds, and the sum of a UAV's path costs cannot exceed this capacity. We use the velocity-ramp model to estimate the time for an agent to travel between points, with $v_{max} = 3\,\text{m/s}$ and $a_{max} = 1\,\text{m/s}^2$ in all experiments. Furthermore, the agents use a fully connected network topology for communication, though this is not always possible in search and rescue missions, the effects of network topology and inconsistent situational awareness have shown to be minimal [13]. To guarantee convergence, we use a maximal assignment factor $\omega = 15$, which avoids auctioning the same tasks repeatedly in certain situations as proposed in [26].

The algorithms and software developed in this study were implemented in Python, and the experiments were performed on a workstation with an Intel Core i5-9400F 2.90 GHz CPU and 16 GB of RAM running Ubuntu 20.04.

We compare our approach of decentralized trajectory task allocation against a state-of-the-art centralized algorithm [11]

using the AC300 dataset [8] to assess performance and potential overhead resulting from using our decentralized method. The AC300 dataset contains 300 realistic outdoor environments. This dataset is based on the EPFL aerial rooftop dataset [29] to provide coverage areas with realistic environments including obstacles. The dataset consists of 300 environments each with a total area of $10,000\,\mathrm{m}^2$ and containing between 0 and 15 obstacles. The aerial robot used in the simulation has a $3\,\mathrm{m}^2$ field-of-view, which results in between 35 to 161 trajectory tasks generated for sufficient coverage. The trajectory tasks are generated by decomposing the polygon including the obstacles into sub-polygons and generating a boustrophedon (lawn-mower) coverage pattern for each sub-polygon [11]. We use the coverage lines generated in [11] and convert them to trajectory tasks for our decentralized task allocation algorithm to enable a fair comparison.

The sum of the agent route costs, in seconds, is evaluated for all the environments and summarized as a total duration for the dataset for both methods using two agents with 1200 s capacity each. The centralized method achieved a total duration of 544592 s and our method achieved 554920 s, and shows that our decentralized trajectory task generalization and methods of estimating the costs only use 10328 s more time resulting in an overhead cost of 1.9% using multiple UAVs when compared to the state-of-the-art centralized area-coverage planning method.

A. Point tasks vs. trajectory tasks

To assess the efficacy and scalability of our solution for multi-UAV coverage planning in realistic environments, we use the AC300 dataset [8]. The initial position, the *depot*, is randomised for all the problems and is common for all agents, to ensure an unbiased comparison with our solution and the current state-of-the-art centralized methods, which uses a common depot for all agents. This is not a constraint and decentralized planning can be conducted regarding the initial positions of the agents. The original datasets are available in [30], and we have made our datasets with the extracted trajectory tasks publicly available in [31].

The impact of our decentralized task allocation algorithm is validated by comparing two different types of scoring functions, (i) a single point cost estimation, by calculating the Euclidean distance to the entry point of the trajectory task, and (ii) optimizing the direction of the task and choosing the lowest-cost direction of travel on the task. We conducted the experiments considering two agents, each with a 1200 s capacity. The comparison of the total travel cost between the two different methods is shown in Fig 2. As can be seen in the figure, our trajectory task approach generates lower-cost routes over different numbers of UAVs, with an average improvement of 6% over the point estimation method.

It is important to note that the performance improvement of the trajectory task estimation method is not the only improvement to the decentralized coverage planning problem. When allowing permutation of the direction for a coverage task, it is possible to generate routes that are intuitive for a human operator. In Fig 3, examples of routes generated using the two different evaluation methods are shown, and here it is clear that the routes calculated using our method are simpler and more intuitive. Simplicity is paramount for the situational awareness of the operator in a SAR mission and might be a critical factor for an operator to accept an autonomous system [4].

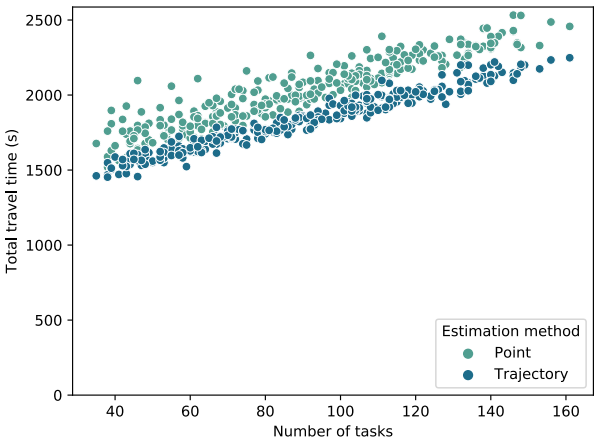


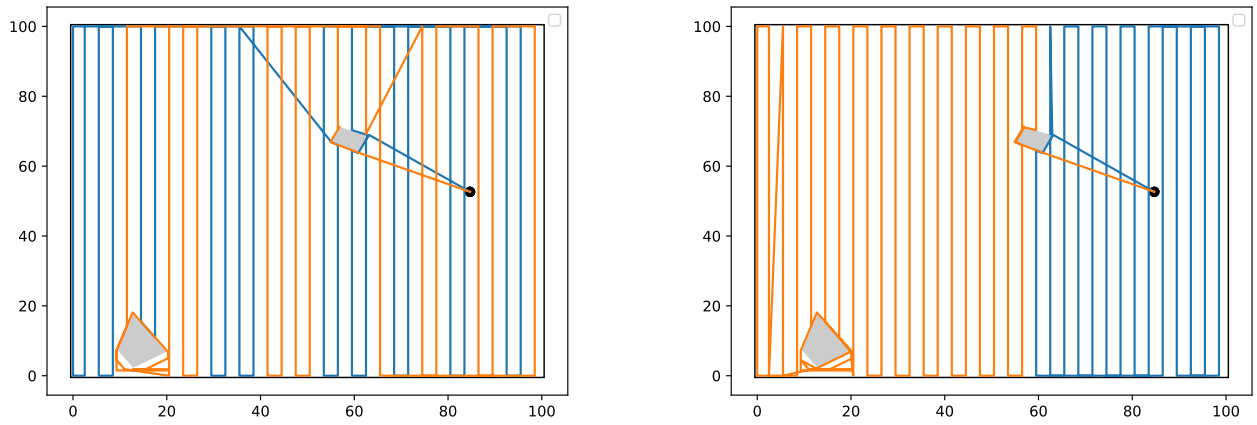
Fig. 2: Shows a comparison of the total travel time of all routes computed using the point scoring function and trajectory scoring function. The results are computed for two UAVs, each with a capacity of 1200 s.

To test the scalability of the proposed approach, we ran the trajectory task allocation algorithm on multi-UAV systems consisting of 2 to 14 agents, see Fig. 4. As can be seen in the figure, the time to completion is significantly improved when adding more UAVs, with an average maximum travel time of 972 s for two UAVs, down to an average travel time of 229 s when 14 UAVs are used. However, the reduction in time to completion of adding more agents stagnates at around 10 UAVs and there is no significant improvement using more than 10 units. The optimal number of UAVs will vary based on the size of the environment.

B. Convergence

The convergence and stability of the system are critical in a SAR operation, therefore it is important to ensure that the algorithm converges within a reasonable timeframe while still providing good solutions. We conducted a set of experiments to determine the number of iterations required to converge for different numbers of agents, see Fig. 5. As can be seen in the figure, the trajectory planning algorithm scales efficiently with the number of agents. There is only a slight increase in iterations when the number of agents is increased, which means that the trajectory task allocation algorithm scales well.

The computational complexity of the trajectory algorithm shares the same computational complexity as CBBA for the bundle construction, see Eq. (3). This means that if the capacity of the agents is decreasing proportionally to the



(a) Point estimation

(b) Trajectory estimation

Fig. 3: Shows a comparison of example routes generated by using (a) the point estimation method, and (b) the trajectory estimation method using an environment with the size of 100×100m from the AC300 dataset. The coverage plans were generated by two UAVs with a capacity of 1200 s, whose paths are orange and blue, respectively.

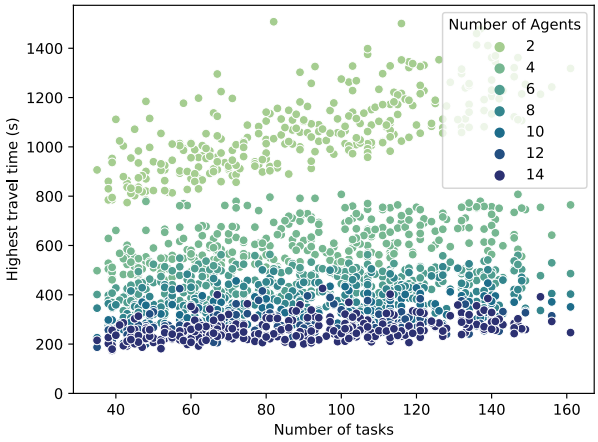


Fig. 4: Comparison between the highest travel time for a single agent in the AC300 problems, across 2 to 14 agents in steps of 2 agents with the capacity constraints set to 1500 s, 750 s, 600 s and the remaining to 500 s, respectively.

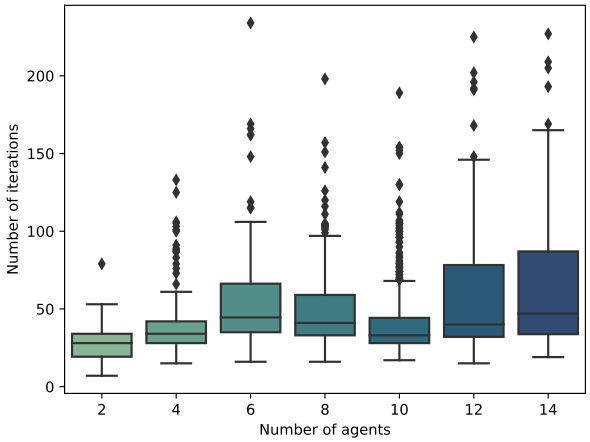


Fig. 5: Comparison between the different number of agents and their respective iterations to convergence using the AC300 dataset across 2 to 14 agents in steps of 2 agents with the capacity constraints set to 1500 s, 750 s, 600 s and the remaining to 500 s, respectively, for all environments in the AC300 dataset. Diamonds represent outliers.

increase in the number of agents, the computation time of constructing the bundle does not increase. As a result, the proposed algorithm converges quickly for multi-robot scenarios containing many capacity-constrained robots, and as the results show, decentralization comes with low overhead when compared to state-of-the-art centralized coverage planning algorithms.

VI. Conclusions

Search and rescue scenarios are time-critical operations, and robust, effective task allocation is crucial when multi-UAV systems are used. In this paper, we presented a novel approach to decentralized task allocation for capacity-constrained robots for coverage planning problems. We generalized the notion of *tasks* to include trajectories, taking into account the spatial nature of coverage planning in the decentralized marked-based task allocation algorithm CBBA. The results show that (i) the approach presented in this paper is capable of generating coverage plans that are competitive with state-of-the-art centralized planners despite being decentralized, and (ii) that it scales well with system size.

In this study, we considered that each trajectory task was a line segment that could be travelled in two different directions. However, a trajectory task can, in principle, be a collection of several line segments or even some non-linear path. The number of tasks that are auctioned during the task allocation process can thus be adjusted based on the available resources: if a search area is very large and a large number of line segments must be covered, each trajectory task could be a cluster of line segments; while if an area is small, more fine-grained trajectory tasks could be defined. In future work, we will investigate clustering line segments to optimize the task allocation process and study dynamic scenarios where UAVs may become available or unavailable during a mission. Our long-term objective is to showcase the efficacy of the proposed methods within a realistic simulation environment that systematically considers dynamic and volatile communication conditions, along with physical constraints. This environment will facilitate the deployment of our proposed methods on physical multi-UAV systems in the field.

REFERENCES

[1] J. P. Queralta, J. Taipalmaa, B. C. Pullinen, V. K. Sarker, T. N. Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund, "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision," *IEEE Access*, vol. 8, pp. 191 617–191 643, 2020.

[2] A. L. Christensen, K. A. R. Grøntved, M.-T. O. Hoang, N. van Berkel, M. Skov, A. Scovill, G. Edwards, K. R. Geipel, L. Dalgaard, U. P. S. Lundquist *et al.*, "The HERD project: Human-multi-robot interaction in search & rescue and in farming," in *Adjunct Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, pp. 1–4.

[3] T. M. Cabreira, L. B. Brisolara, and F. J. Paulo R, "Survey on coverage path planning with unmanned aerial vehicles," *Drones*, vol. 3, no. 1, p. 4, 2019.

[4] M.-T. Hoang, N. van Berkel, K. A. R. Grøntved, M. Skov, A. L. Christensen, and T. Merritt, "Drone Swarms to Support Search and Rescue Operations: Opportunities and Challenges," in *Cultural Robotics: Social Robots and their Emergent Cultural Ecologies*, B. J. Dunstan, J. T. K. V. Koh, D. T. Tillman, and S. A. Brown, Eds. Springer, 2023.

[5] B. O. Koopman, "The theory of search: III. The optimum distribution of searching effort," *Operations Research*, vol. 5, no. 5, pp. 613–626, 1957.

[6] A. L. Adams, T. A. Schmidt, C. D. Newgard, C. S. Federiuk, M. Christie, S. Scorvo, and M. DeFreest, "Search is a time-critical event: when search and rescue missions may become futile," *Wilderness & Environmental Medicine*, vol. 18, no. 2, pp. 95–101, 2007.

[7] H. Choset and P. Pignon, "Coverage path planning: The boustrophedon cellular decomposition," in *Field and Service Robotics*. Springer, 1998, pp. 203–209.

[8] R. Bähnemann, N. Lawrance, J. J. Chung, M. Pantic, R. Siegwart, and J. Nieto, "Revisiting boustrophedon coverage path planning as a generalized traveling salesman problem," in *Field and Service Robotics*. Springer, 2021, pp. 277–290.

[9] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[10] S. Agarwal and S. Akella, "Line coverage with multiple robots," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3248–3254.

[11] ——, "Area coverage with multiple capacity-constrained robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3734–3741, 2022.

[12] M. G. Lagoudakis, M. Berhault, S. Koenig, P. Keskinocak, and A. J. Kleywegt, "Simple auctions with performance guarantees for multi-robot task allocation," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1. IEEE, 2004, pp. 698–705.

[13] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 912–926, 2009.

[14] I. Maza and A. Ollero, "Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms," in *Distributed Autonomous Robotic Systems 6*. Springer, 2007, pp. 221–230.

[15] J. I. Vasquez-Gomez, J.-C. Herrera-Lozada, and M. Olguin-Carbajal, "Coverage path planning for surveying disjoint areas," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2018, pp. 899–904.

[16] E. M. Arkin, S. P. Fekete, and J. S. Mitchell, "Approximation algorithms for lawn mowing and milling," *Computational Geometry*, vol. 17, no. 1-2, pp. 25–50, 2000.

[17] N. Jozefowiez, F. Semet, and E.-G. Talbi, "Multi-objective vehicle routing problems," *European Journal of Operational Research*, vol. 189, no. 2, pp. 293–309, 2008.

[18] I. Vandermeulen, R. Groß, and A. Kolling, "Turn-minimizing multi-robot coverage," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 1014–1020.

[19] A. R. Mosteo, L. Montano, and M. G. Lagoudakis, "Multi-robot routing under limited communication range," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 1531–1536.

[20] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination: A survey and analysis," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1257–1270, 2006.

[21] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 427–438, 2012.

[22] M. Berhault, H. Huang, P. Keskinocak, S. Koenig, W. Elmaghraby, P. Griffin, and A. Kleywegt, "Robot exploration with combinatorial auctions," in *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2. IEEE, 2003, pp. 1957–1962.

[23] L. Johnson, S. Ponda, H.-L. Choi, and J. How, "Asynchronous decentralized task allocation for dynamic environments," in *Infotech@ Aerospace 2011*, 2011, p. 1441.

[24] H.-L. Choi, A. K. Whitten, and J. P. How, "Decentralized task allocation for heterogeneous teams with cooperation constraints," in *Proceedings of the 2010 American Control Conference*. IEEE, 2010, pp. 3057–3062.

[25] A. Whitbrook, Q. Meng, and P. W. Chung, "A novel distributed scheduling algorithm for time-critical multi-agent systems," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 6451–6458.

[26] J. Turner, Q. Meng, G. Schaefer, A. Whitbrook, and A. Soltoggio, "Distributed task rescheduling with time constraints for the optimization of total task allocations in a multirobot system," *IEEE Transactions on Cybernetics*, vol. 48, no. 9, pp. 2583–2597, 2017.

[27] J. Tang, K. Zhu, H. Guo, C. Gong, C. Liao, and S. Zhang, "Using auction-based task allocation scheme for simulation optimization of search and rescue in disaster relief," *Simulation Modelling Practice and Theory*, vol. 82, pp. 132–146, 2018.

[28] W. Zhao, Q. Meng, and P. W. Chung, "A heuristic distributed task allocation method for multivehicle multitask problems and its application to search and rescue scenario," *IEEE Transactions on Cybernetics*, vol. 46, no. 4, pp. 902–915, 2015.

[29] X. Sun, C. M. Christoudias, and P. Fua, "Free-shape polygonal object localization," in *13th European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 317–332.

[30] S. Agarwal and S. Akella, "AreaCoverage-dataset," https://github.com/UNCCharlotte-CS-Robotics/AreaCoverage-dataset, 2021, [Online; accessed 1-March-2023].

[31] K. A. R. Grøntved, U. P. S. Lundquist, and A. L. Christensen, "Area coverage task dataset for multi-robot task allocation," https://github.com/kasperg3/CoverageTasks, 2023.