Scheduling drones to cover outdoor events

O. Aichholzer[1], L. E. Caraballo[2], J.M. Díaz-Báñez[2], R. Fabila-Monroy[3], I. Parada[1], I. Ventura[2], and B. Vogtenhuber[1]

1 **Graz University of Technology**
`oaich@ist.tugraz.at,iparada@ist.tugraz.at,bvogt@ist.tugraz.at`
2 **University of Seville**
`lcaraballo@us.es,dbanez@us.es,iventura@us.es`
3 **Cinvestav**
`ruyfabila@math.cinvestav.edu.mx`

Abstract

Task allocation is an important aspect of many multi-robot systems. In this paper, we consider a new task allocation problem that appears in multi-robot aerial cinematography. The main goal is to distribute a set of tasks (shooting actions) among the team members optimizing a certain objective function. The tasks are given as sequences of waypoints with associated time intervals (scenes). We prove that the task allocation problem maximizing the total filmed time by k aerial robots (drones) can be solved in polynomial time when the drones do not require battery recharge. We also consider the problem in which the drones have a limited battery endurance and must periodically go to a static base station. For this version, we show how to solve the problem in polynomial time when only one drone is available.

1 Introduction

In the last twenty years, the use of aerial robots and aerial multi-robot systems in monitoring, surveillance, delivery of goods, network coverage, etc. [8], has brought several challenges that have attracted the attention of both, robotics and mathematics research communities. For example, the Vehicle Routing Problem (VRP) and Traveling Salesman Problem (TSP) are classical problems in the area of operations research that have got renewed attention with these applications; see [1] for a survey on VRP instances with applications to multi-objective unmanned aerial vehicle operations and [2, 4, 5] for studies in commercial scenarios that consider a combination of trucks and drones to perform the so-called last-mile delivery. Also, task allocation problems have been widely addressed in the interplay of aerial robotics and operations research. See [7] for a review on multi-robot task allocation.

The problems studied in this paper are inspired by the use of unmanned aerial vehicles (drones, also called UAVs) for autonomous cinematography planning, aimed at filming outdoor events (e.g., sport events such as cycling); see [9]. Drones are ideal to cover outdoor events in large spaces, as they do not require the existence of previous infrastructure and they can operate in places of difficult access. The input is provided by the media production director, who specifies multiple shots that should be executed to film different scenes of the event. These are basically time intervals associated with waypoints on the terrain. Each shot may require one or multiple cameras, but in the end, all information can be translated into a set

This research has been supported by the projects GALGO (Spanish Ministry of Economy and Competitiveness and MTM2016-76272-R AEI/FEDER,UE, MULTIDRONE, Grant Agreement number: 731667-H2020-ICT-2016-2017), by the Austrian Science Fund within the collaborative DACH project *Arrangements and Drawings* as FWF project I 3340-N35, and by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 734922.

36th European Workshop on Computational Geometry, Würzburg, Germany, March 16–18, 2020.
This is an extended abstract of a presentation given at EuroCG'20. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

of filming tasks with temporal and spatial constraints to be accomplished by the team. The goal is to schedule the drones optimally in order to cover the filming tasks.

Problem statement

Consider a scenario in which an outdoor event is filmed by a set of drones. In the model we assume that all drones fly at unit speed and start at a point p^* in the plane called the *base*. The production director specifies certain locations and time intervals at which the filming of certain scenes is desired. We represent the input as a set F of n tuples (p_i, I_i) with $i \in \{1, \dots, n\}$, where p_i is a point in space and I_i is a time interval. We call F the *film plan* and each (p_i, I_i) a *scene*. A scene (p_i, I_i), or part of it, can be filmed by one or multiple drones located at p_i during (part of) the time interval I_i. A *flight plan* P is a sequence of tuples $(q_1, J_1), \dots, (q_m, J_m)$ such that for every j, q_j is equal to some p_i and $J_j \subseteq I_i$. The flight plan is assumed to be realizable by a drone starting at the base. The goal is to assign flight plans to all drones in order to film as much as possible of the film plan. The *filming time* of a flight plan assignment M is the sum of the lengths of the subintervals of I_i covered by the flight plans. Formally, it is defined as

$$\sum_{i=1}^{n} \left| \bigcup_{\substack{P \in M \\ (p_i, J_j) \in P}} (I_i \cap J_j) \right| .$$

In this paper we study the following algorithmic problems in the above scenario.

▶ **Problem 1.** *Find a flight plan maximizing the filming time for a single drone.*

▶ **Problem 2.** *Find a flight plan assignment maximizing the filming time for $k \geq 2$ drones.*

▶ **Problem 3.** *Find the minimum number of drones needed to capture the complete film plan, plus an according flight plan assignment.*

Typically, drones have limited battery endurance and periodically return to a base station to recharge or change their batteries. Let $L > 0$ be a real number and assume that each drone can fly for time at most L before it must be at the base for a battery change. We call L the *battery life*. In our model, a battery change is assumed to be instantaneous and the drone can resume its flight plan immediately after arriving at the base. The drone can also wait at the base for an arbitrary amount of time without consuming its battery.

Results

We present the following results. In the case of unlimited battery life, Problem 1 can be solved in $O(n^{5/3} + |E|)$ time and $O(|E|)$ space for n time intervals. Here, $|E|$ is the size of a graph that in the worst case is quadratic in n but it is linear in realistic inputs for cycling events [9]. For the general case in which k drones are available, we show how to solve Problem 2 in $O(n^2(\log n + k) + n|E|)$ time, while Problem 3 can be solved in $O(n^{5/3} + \sqrt{n}|E'|)$ time, where $|E'| < |E|$. All according proofs are constructive in the sense that they also provide a flight plan that attains the computed maximum time/minimum number of drones.

The case of limited battery life is more challenging. Note that the complexity of an explicit flight plan not only depends on the number of intervals, but also on the ratio between the total time T of the film plan and the battery life L. We show that Problem 1 can still be solved in polynomial time by augmenting the drone's model so that repeated identical instructions can be formulated in a compact way. We conjecture the general case for k drones with limited battery life to be NP-hard.

2 Unlimited battery life

In this section we consider the scenario where the drones do not require to recharge their batteries. We first state a lemma that allows us to discretize the solution space for Problems 1, 2, and 3 in this setting. The intuition behind Lemma 2.1 is the following. In an optimal solution, there is no reason for a drone at a point p_i to leave its current filming position before the corresponding interval I_i ends. Moreover, there is no reason for two drones to be filming at the same location at the same time. The proof of Lemma 2.1 is technical and it is based on applying the following two operations on flight plans.

Let $P := (q_1, J_1), \ldots, (q_m, J_m)$ be a flight plan, t a real number, and $1 \leq i \leq m$. Informally, our first operation yields the flight plan obtained from P by staying additional time t at position q_i and thus arriving at the subsequent intervals at later times. We formalize this definition. For a time interval $[a, b]$ we define

$$\begin{aligned} \text{extend}([a,b],t) &:= [a, b+t] \\ \text{delay}([a,b],t) &:= \begin{cases} [a,b] & \text{if } t \leq 0 \\ [a+t,b] & \text{if } 0 < t < b-a, \\ \emptyset & \text{if } t \geq b-a. \end{cases} \end{aligned}$$

Consider the sequence $P' := (q_1, J_1), \ldots, (q_{i-1}, J_{i-1}), (q_i, \text{extend}(J_i, t)), (q_{i+1}, \text{delay}(J_{i+1}, t)),$ $(q_{i+2}, \text{delay}(J_{i+2}, t - |J_{i+1}|)), (q_{i+3}, \text{delay}(J_{i+3}, t - |J_{i+1}| - |J_{i+2}|)), \ldots, (q_m, \text{delay}(J_m, t - \sum_{j=i+1}^{m-1} |J_j|))$. Let P'' be the flight plan obtained from P' by removing every tuple whose time interval is empty. We call P'' the flight plan obtained from P by *shifting P at interval J_i by time t*; see Figure 1.

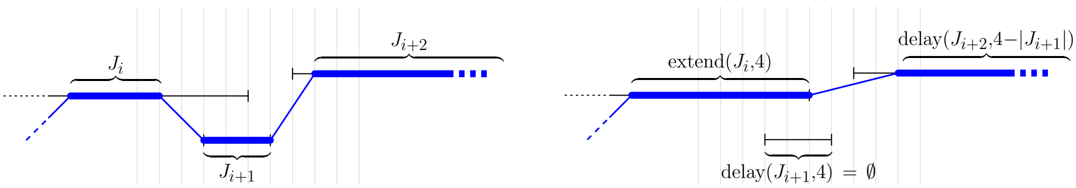

Figure 1 Example of a shifting operation.

Our second operation involves two drones that meet at some point p_i at the same time. Informally, the operation makes the drones swap their flight plans at the point of contact. Let $P_1 := (q_1, J_1), \ldots, (q_m, J_m)$ and $P_2 := (r_1, K_1), \ldots, (r_s, K_s)$ be two flight plans such that for some pair of indices $1 \leq i \leq m$ and $1 \leq j \leq s$ the following holds. The point q_i equals the point r_j and the last point of interval $J_i := [a, b]$ is contained in the interval $K_j := [c, d]$. We call the operation of replacing P_1 and P_2 with the flight plans

$$P_1' := \begin{cases} (q_1, J_1), \ldots, (q_{i-1}, J_{i-1}), (q_i, [a, d]), (r_{j+1}, K_{j+1}), \ldots, (r_s, K_s) & \text{if } a < c, \\ (q_1, J_1), \ldots, (q_{i-1}, J_{i-1}), (q_{i+1}, J_{i+1}), \ldots, (q_m, J_m) & \text{if } a \geq c. \end{cases}$$

$$P_2' := \begin{cases} (r_1, K_1), \ldots, (r_{j-1}, K_{j-1}), (q_{i+1}, J_{i+1}), \ldots, (q_m, J_m) & \text{if } a < c, \\ P_2 & \text{if } a \geq c. \end{cases}$$

swapping P_1 with P_2 at interval J_i; see Figure 2.

Note that any shifting or swapping operation maintains both realizability and total filming time of the involved flight plan(s).

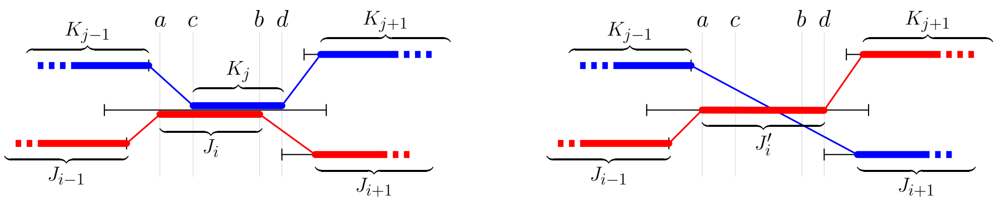

Figure 2 Example of a swapping operation.

▶ **Lemma 2.1.** *For every film plan there exists a flight plan of maximum filming time where*
1) *no drone leaves a point p_i before the interval I_i ends; and*
2) *no two drones are at the same point p_i at the same time.*

By Lemma 2.1, we may assume that in an optimal solution to Problems 1, 2, or 3 a drone never leaves a point p_i before the interval I_i ends. We can use this property to translate these problems into problems of covering directed weighted acyclic graphs with directed paths. We construct a directed graph $G = (V, E)$. The vertex set V consists of p^* and the points p_i. A pair (p_i, p_j) is an edge in the edge set E whenever a drone at a point p_i can leave at the end of I_i and arrive at p_j at a time $t \in I_j := [a, b]$. In such a case, (p_i, p_j) is assigned weight $b - t$. Every pair (p^*, p_i) is an edge in E with weight $|I_i|$. Due to geographic and time constrains, not all pairs (p_i, p_j) might be edges of E. We can compute E efficiently, in an output sensitive manner.

▶ **Lemma 2.2.** *The edge set E can be computed in $O(n^{5/3} + |E|)$ time.*

By Lemma 2.1, a flight plan for one drone that maximizes the filming time corresponds to a directed path starting at p^* of maximum weight. Since G is acyclic, such a path can be computed in $O(|E| + n)$ time by first doing a topological sort of G and then finding the desired path via dynamic programming. This solves Problem 1.

▶ **Theorem 2.3.** *Problem 1 with unlimited battery life can be solved in $O(n^{5/3} + |E|)$ time.*

Suppose that $k \geq 2$ drones are available. By Lemma 2.1, a flight plan assignment for these drones that maximizes the filming time corresponds to a set of k internally disjoint paths of G, all starting starting at p^* maximizing the sum of its weights. It is known that the problem of finding k disjoint paths of maximum total weight all starting at a given vertex in a general directed graph is NP-complete [6]. However, since G is acyclic, we can prove the following result.

▶ **Theorem 2.4.** *Problem 2 with unlimited battery life can be solved in $O(n^2(\log n + k) + n|E|)$ time.*

Now, let $G' = (V, E')$ be the following subgraph of G. The set E' is the subset of E of edges (p^*, p_j) and the edges (p_i, p_j) of weight equal to $|I_j|$. This corresponds to the cases in which a drone at p_i leaving when I_i ends can arrive at p_j just when I_j starts. In a similar way to Lemma 2.2, E' can be computed in $O(n^{5/4} + |E'|)$ time. By Lemma 2.1, an optimal solution to Problem 3 corresponds to a set of minimum cardinality of vertex disjoint paths that covers every vertex of G'. Such a set is called a *minimum path cover*. The problem of finding a minimum path cover for general directed graphs is NP-hard. However, if the digraph is acyclic then a minimum path cover can be computed in $O(\sqrt{n}m)$ time [3], where n is the number of vertices and m is the number of edges of the digraph.

▶ **Theorem 2.5.** *Problem 3 with unlimited battery life can be solved in $O(n^{5/3} + \sqrt{n}|E'|)$ time.*

3 Limited battery life for one drone

With bounded battery for each drone, the problem of selecting and scheduling drones optimally to shoot the provided time intervals is a challenging problem. It is not difficult to build an example for two drones in which, for an optimal solution, the leaving time for a drone is not necessarily the end of an interval. We conjecture the problem is NP-hard in general. In this section, we consider the problem for one drone and battery life L. This implies that the drone must return to the base at latest after time L for a battery change. We refer to the subsequence of a flight plan between two consecutive visits to the base as a *round*. Let $T = |\bigcup_{i=1}^{n} I_i|$. Note that the number of rounds of any flight plan of maximum filming time for one drone is at least $\lfloor T/L \rfloor$ (it could be exponential in n). As in the unlimited battery case, we can discretize the solution space.

▶ **Lemma 3.1.** *For every film plan there exists a flight plan of maximum filming time for one drone with limited battery life such that:*

1) If (p_j, J_i) and (p'_j, J_{i+1}) are consecutive tuples in the flight plan, then the last point of J_i is equal to the last point of I_i.

2) If (p^*, J_i) and (p_j, J_{i+1}) are consecutive tuples in the flight plan and $|J_i| > 0$, then the first point of J_i is equal to the first point of I_j

3) If (p_j, J_i) and (p^*, J_{i+1}) are consecutive tuples in the flight plan, then the last point of J_i is equal to the last point of I_j or the drone is running out of battery at the last point of J_i.

Lemma 3.1 implies that the set of theoretically relevant event-times for some optimal solution is discrete rather than continuous. Figure 3 shows an example scenario with an optimal flight plan.

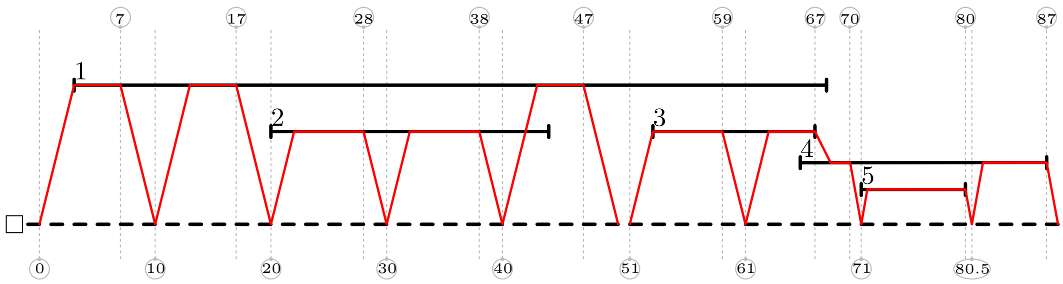

Figure 3 Scenario with 5 scenes represented by solid black lines. The dashed black line represents the base. The red solid stroke represents the movement of the drone following an optimal solution. The battery lasts 10 units of time. Closed in gray circles are the time instants associated to the drone's moves.

We can show that the problem can be solved in polynomial time if we augment the computational model of the drone. More precisely, our drone operates from a set of *driving instructions* of the type "go to the base" and "go to interval I_i". In the augmented model, the drone acts by default according to an internal protocol that allows the drone to perform several moves between two consecutive instructions. In this model, "go to interval I_i" means "repeatedly go to interval I_i, film as long as possible, and return to the base station, until the next instruction comes". Thus, an optimal scheduling can be encoded by a polynomial sequence of driving instructions. A crucial result is the following.

▶ **Lemma 3.2.** *An optimal solution for Problem 1 with limited battery life can be encoded using a linear number of driving instructions.*

The results above allow us to apply dynamic programming to compute an optimal sequence of instructions in polynomial time.

▶ **Theorem 3.3.** *Problem 1 with limited battery life can be solved in polynomial time.*

References

1 M. Adbelhafiz, A. Mostafa, and A. Girard. Vehicle routing problem instances: Application to multi-UAV mission planning. In *AIAA Guidance, Navigation, and Control Conference*, 2010. doi:10.2514/6.2010-8435.

2 N. Agatz, P. Bouman, and M. Schmidt. Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, 52(4):965–981, 2018. doi:10.1287/trsc.2017.0791.

3 F. T. Boesch and J. F. Gimpel. Covering points of a digraph with point-disjoint paths and its application to code optimization. *Journal of the ACM*, 24(2):192–198, 1977. doi:10.1145/322003.322005.

4 John Gunnar Carlsson and Siyuan Song. Coordinated logistics with a truck and a drone. *Management Science*, 64(9):4052–4069, 2018. doi:10.1287/mnsc.2017.2824.

5 C. C. Murray and A. G. Chu. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, 54:86–109, 2015. doi:10.1016/j.trc.2015.03.005.

6 G. Naves, N. Sonnerat, and A. Vetta. Maximum flows on disjoint paths. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques. 13th International Workshop*, pages 326–337, 2010. doi:10.1007/978-3-642-15369-3_25.

7 E. Nunes, M. Manner, H. Mitiche, and M. Gini. A taxonomy for task allocation problems with temporal and ordering constraints. *Robotics and Autonomous Systems*, 90:55–70, 2017. doi:10.1016/j.robot.2016.10.008.

8 H. Shakhatreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani. Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges. *IEEE Access*, 7:48572–48634, 2019. doi:10.1109/ACCESS.2019.2909530.

9 A. Torres-González, J. Capitán, R. Cunha, A. Ollero, and I. Mademlis. A multidrone approach for autonomous cinematography planning. In *Iberian Robotics Conference*, pages 337–349, 2017.