Covering line segments with drones: the minmax criterion*

José-Miguel Díaz-Bañez, José Manuel Higes, Alina Kasiuk, and Inmaculada Ventura

Departamento de Matemática Aplicada II, Universidad de Sevilla
{dbanez, jhiges, akasiuk, iventura}@us.es

Abstract

We are given a set of line segments (e.g., tubes in a solar plant) to be inspected by drones. The limited capacity of the batteries imposes periodical visits (tours) to a fixed base station. The objective is to assign a set of tours for each drone so that the segments are covered as quickly as possible, i.e. to minimize the maximum time spent by the drones. In this paper, we prove that this problem is NP-hard even when the segments are positioned on a line and the scenario involves two drones. An approximation algorithm is proposed with constant factor ranging form 1 to 2.

1 Introduction

As technology advances, unmanned aerial vehicles (UAVs), commonly referred to as drones, are assuming an ever-expanding role in the inspection of industrial structures. For example, the manual inspection of high-voltage power transmission lines or solar plants are both time-consuming and expensive [6]. Hence, the use of drones equipped with cameras enables efficient fault detection.

In the case of Concentrated Solar Power plants (CSP), which represent a growing technology for electricity production through renewable energies, the plant comprises an array of receiver tubes subjected to high thermal stress. Therefore, the identification of broken glass envelopes is crucial to maintain the proper functioning of the CSP plant [5]. In this context, promptly detecting a fault enables the company to take immediate action in the repair process. In this paper, we tackle the problem of minimizing the time required for a team of drones to effectively traverse a set of tubes, represented as line segments. This problem falls within the domain of arc routing problems (ARPs), which involve determining a set of tours with the minimum total cost while traversing a set of links (arcs or edges) known as required links in a graph [4]. However, in contrast to vehicles in classical ARPs, a drone has the flexibility to enter a line through any of its points, traverse a portion of that line, exit through another of its points, and subsequently travel directly to any point on another line, and so forth. Hence, employing drones for service in ARPs introduces substantial modifications to the conventional methods of modeling and solving these problems [2].

In a recent paper [1], the authors study a problem focused on minimizing the total time required for one drone to cover a set of line segments positioned along a given line. Here, the total time is defined as the sum of the lengths of the necessary tours. The drone has limited battery endurance, maintains a constant flying speed, and returns to a base station when its battery is running low. They show that this one-dimensional variant can be solved in polynomial time. They also address the problem of minimizing the number of tours needed

* This work is partially supported by grants PID2020-114154RB-I00 and TED2021-129182B-I00 funded by MCIN/AEI/10.13039/501100011033 and the European Union NextGenerationEU/PRTR.

40th European Workshop on Computational Geometry, Ioannina, Greece, March 13–15, 2024.
This is an extended abstract of a presentation given at EuroCG'24. 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.

to cover all the segments. In this paper, we consider the same one-dimensional covering problem, with a focus on minimizing the maximum distance (time) traveled by the drones of the team—essentially, reducing the time required for the team to cover the segments. This objective is meaningful when the company aims to expedite the task and promptly repair the broken tubes.

1.1 Problem Statement

Let $\alpha = \{a_1, a_2, \cdots, a_n\}$ be a set of disjoint line segments on a line. For $i = 1, 2, \cdots, n$, $a_i = [x_i, y_i]$, with $x_1 < y_1 < x_2 < y_2 < \cdots < x_n < y_n$. We are given a team of k identical drones that must traverse all the segments. The drones are constrained by finite battery endurance, maintain uniform velocity during flight, and execute takeoff and battery recharging protocols at a fixed point B, the designated base station, situated exterior to the line. We assume that the recharging time is negligible as the batteries are replaced instantly. Let $L > 0$ denote the maximum distance achievable by a drone when initiating and concluding its trajectory (tour) at the fixed point B. For a tour labeled as t, we refer to its length as $\ell(t)$. The length of a collection of tours, denoted as $T = \{t_1, \cdots, t_m\}$, is represented as $\ell(T)$. It is calculated as the sum of the lengths of individual tours, that is, $\sum_{i=1}^{m} \ell(t_i)$. Our objective is to determine a set of tours for each drone in such a way that we minimize the maximum length traveled by any drone (the makespan) while ensuring that all segments are covered. Formally, the *Minmax problem* for k drones can be stated as follows:

▶ **Problem 1.1.** *Minmax-k: compute a set of tours for each drone, $T_1, T_2, \cdots, T_k$, such that:*

$$\alpha \subset T_1 \cup T_2 \cup \cdots \cup T_k \textit{ and,} \tag{1}$$

$$\max_{j=1,\cdots,k} \ell(T_j) \textit{ is minimized.} \tag{2}$$

See Figure 1 for an example of a set of tours covering line segments with $k = 2$ drones.

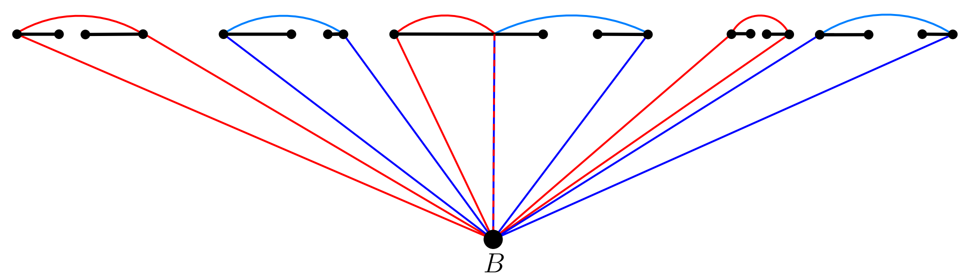

Figure 1 Covering tours for two drones. Blue and red tours correspond to different drones.

The problem of minimizing the maximal number of tours (instead of total length) performed by the k drones can be easily addressed using the approach of [1]. Indeed, let m be the minimum number of tours required by just a drone to cover all segments in the set S. Thus, $\lceil \frac{m}{k} \rceil$ is the solution, and the problem can be solved in $O(n + m)$ time with a greedy approach proposed in [1]. In the same paper, it has been demonstrated that the *Minsum problem*, that is, minimizing the total length of a set of tours performed by a single drone can also be solved in polynomial time. This can be easily extended to solve the Minsum problem for k drones by allowing one drone to perform all the tours while the others remain inactive. In this paper, we prove that transitioning from the Minsum problem to the Minmax criterion for two drones results in an NP-hard problem. Then, an approximation algorithm with factor ranging from 1 to 2 is proposed.

2 NP-Hardness

It is known that the *two way-balanced partition problem* is NP-complete [3]: Given a finite set of positive integers $S = \{s_i\}_{i=1}^n$, determine if there is a subset $A \subset S$ of cardinality $\lfloor n/2 \rfloor$, with $\sum_{s_i \in A} s_i = \sum_{s_j \in S \setminus A} s_j$.

We can prove that the following variant is NP-hard:

▶ **Problem 2.1.** *Minmax partition problem: Given a finite set of positive integers $S = \{s_i\}_{i=1}^{2n}$, determine a subset $A \subset S$ of cardinality n, with $M_2 = \sum_{s_i \in A} s_i \geq M_1 = \sum_{s_j \in S \setminus A} s_j$, such that M_2 is minimum for all possible subsets of S of cardinality n.*

In this section, we outline the key ideas for proving the NP-hardness of the Minmax-2 problem through a reduction from Problem 2.1. The following result will be one crucial tool in our construction:

▶ **Proposition 2.2.** *For a set of positive numbers $S = \{s_i\}_{i=1}^{2n}$ and two positive constants $K, C > 0$, define the set $S' = \{s'_i\}_{i=1}^{2n}$ with $s'_i = Ks_i + C$. Then a subset $A \subset S$ is a solution to Problem 2.1 for S if and only if the subset A', defined by $s'_i \in A'$ if $s_i \in A$, is a solution to Problem 2.1 for the set S'.*

2.1 Construction

The following construction can be done in polynomial time. Given a line, an exterior point B, and a positive constant $L > 0$, let O be the projection point of the base B on the line located as in Figure 2. Let $\{s_i\}_{i=1}^{2n}$ be a set of positive integers and assume w.l.o.g. $R = \max_i\{s_i\} = s_{2n}$. Take ϵ a small number less than 1/3, for example $\epsilon = 10^{-8}$.

- *Step 1:* For $i = 0, \cdots, 2n$, determine a sequence of right hand points y_i, with $c_i = d(y_i, B)$, that satisfies:

$$c_0 > \max\left(\frac{L}{2} - \frac{\epsilon L}{4n}, L - y_0 - \frac{L}{3}, \frac{\sqrt{5}L}{6}\right),$$

$$y_{i+1} - y_i + c_i + c_{i+1} = L,$$

$$c_{i+1}^2 = y_{i+1}^2 + \left(\frac{L}{3}\right)^2.$$

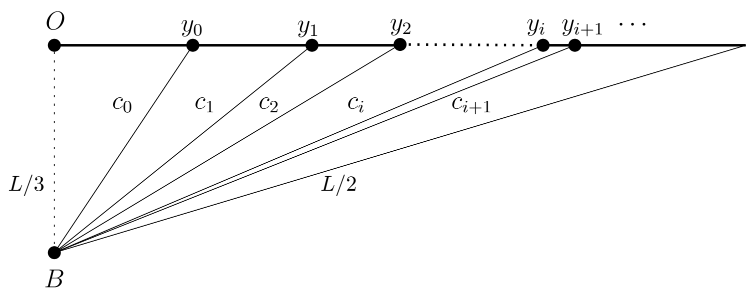

Figure 2 Diagram for Step 1. Right hand points y_i.

- *Step 2:* Determine the number sequence:

$$\{s'_i = Ks_i + 2c_{2n}\}_{i=1}^{2n} \text{ with } K = \frac{L - 2c_{2n}}{R}.$$

- *Step 3:* For $i = 1, \cdots, 2n$, determine the sequence of right hand points x_i, with $b_i = d(x_i, B)$ that satisfies:

$$c_i + (y_i - x_i) + b_i = s'_i,$$

$$b_i^2 = x_i^2 + \left(\frac{L}{3}\right)^2.$$

- *Step 4:* For $i = 1, \cdots, 2n$, draw the subsequence of right hand segments $[x_i, y_i]$ for i even, and left hand segments $[-y_i, -x_i]$ for i odd (Figure 3).

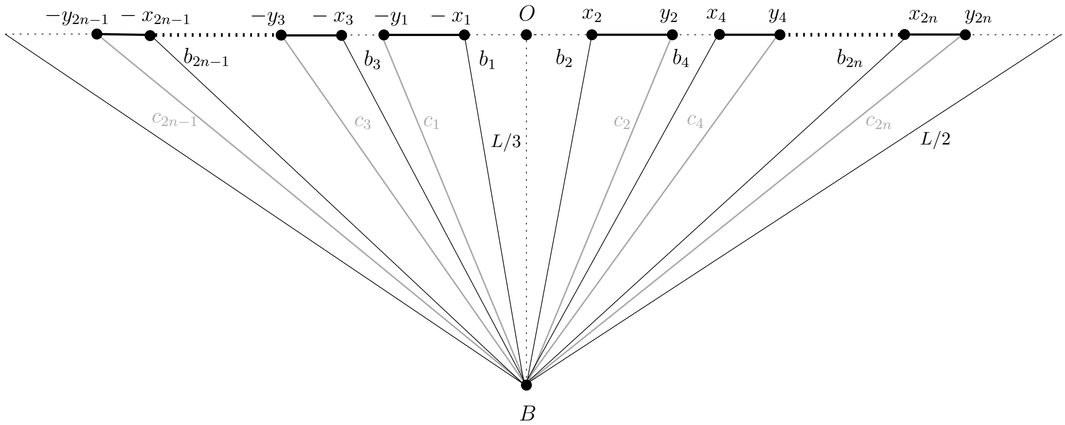

Figure 3 Diagram for step 4. Final construction.

▶ **Theorem 2.3.** *The Minmax-2 problem is NP-hard.*

Proof. We will provide a brief outline of the proof. The problem Minmax-2 is clearly in NP. Given a set of positive integers $S = \{s_i\}_{i=1}^{2n}$, perform the above construction in polynomial time. Set $K = \frac{L-2c_{2n}}{R}$ and $C = 2c_{2n}$. We can prove that the following facts are true for our construction:

- *Statement 1:* The solution of the Minsum problem for one drone in our construction is determined by a the set of tours $T = \{t_i\}_{i=1}^{2n}$ with each tour t_i covering exactly the i-th segment. Notice that the length of each tour t_i is $s'_i = Ks_i + C$ by Step 3.
- *Statement 2:* The tours of any solution $\{T_1, T_2\}$ for the Minmax-2 problem in the construction correspond to tours t_i derived from a solution to the Minsum problem for a single drone.

Now, given a solution A for Problem 2.1 for S, K and C, define the set S' and A' as in Proposition 2.2 and take a solution of the Minsum problem for one drone (by applying the polynomial time algorithm of [1] to our construction). By Statement 1, each tour t_i of this solution covers exactly the i-th segment and has length s'_i. Thus, assigning each tour t_i to T_2 if and only if $s'_i \in A'$, we will have a solution of the Minmax-2 problem by Statement 2, the minimality of A' and the fact that s'_i is the length of the tour t_i.

In the other direction, given a solution for the Minmax-2 problem in our construction, $\{T_1, T_2\}$, assign $s'_i \in A'$ if and only if $t_i \in T_2$ (assume w.o.l.g that $\ell(T_2) \geq \ell(T_1)$). By Statement 2, the length of each t_i is s'_i; therefore we obtain a solution of Problem 2.1 for S' and, by Proposition 2.2, we can consequently derive a solution for S. ◀

3 An approximation algorithm

Greedy tour distribution for two drones, G2D-algorithm: In [1], an algorithm based on dynamic programming to compute a set of tours $T = \{t_1, \cdots, t_m\}$ solving the minsum problem for one drone was proposed. Getting T as the initial step, the G2D-algorithm just distribute the tours of T into two sets, T_1 and T_2, so that, in each step, the difference between the sum of the lengths of the tours in each set is $|\ell(T_2) - \ell(T_1)| = aL$ with $0 \leq a \leq 1$. To do it, given $T = \{t_1, \cdots, t_m\}$, add for $i = 1$ the tour t_1 to T_1 and in each subsequent step $i > 1$ add the tour $t_i \in T$ to the set T_1 or T_2 with minimum total length.

▶ **Observation 3.1.** *Assume w.l.o.g. that $\ell(T_2) \geq \ell(T_1)$ in G2D-algorithm; then, for some $a \in [0, 1]$, $\ell(T_2) = \ell(T_1) + aL$.*

▶ **Theorem 3.2.** *Let $\{T_1^*, T_2^*\}$ be any solution of the Minmax-2 problem and let $\{T_1, T_2\}$ be the final distribution of the G2D-algorithm. Assume $\ell(T_1) \leq \ell(T_2)$ and $\ell(T_1^*) \leq \ell(T_2^*)$. Then:*

a) $\ell(T_1) + \dfrac{aL}{2} \leq \ell(T_2^*) \leq \ell(T_1) + aL$

b) *If $a = 0$, then $\{T_1, T_2\}$ is optimal for the Minmax-2 problem.*

c) *If $a \in (0, 1]$, then $\ell(T_2) \leq \Delta \cdot \ell(T_2^*)$, with $\Delta = \dfrac{\Gamma + 2}{\Gamma + 1}$ and $\Gamma = \dfrac{2\ell(T_1)}{\ell(T_2) - \ell(T_1)}$.*

▶ **Observation 3.3.** *By the results of [1], G2D-algorithm computes T_1 and T_2 in $O(n^2) + O(nm)$ time, where n is the number of segments and m the total number of tours.*

▶ **Observation 3.4.** *As $\ell(T_1)$ increases (for example, if $\ell(T)$ is large) then Δ tends to 1.*

▶ **Observation 3.5.** *G2D-algorithm can be generalized for $k > 2$ drones with the same time complexity; we just have to properly distribute the m tours of T into k sets $T_1, \cdots, T_k$. The approximation factor is $\Delta = \dfrac{\Gamma + k}{\Gamma + 1}$ with $\Gamma = \dfrac{k \cdot \ell(T_1)}{\ell(T_k) - \ell(T_1)}$, where T_1 is the set with minimum total length and T_k the set with maximum total length.*

4 Conclusions and future research

We have proved that covering a set of line segments on a line with drones using the minmax criterion is NP-hard, even when considering only two drones. We found a reduction from a variant of the finite partition problem. It is highly likely that we can extend this result to the Minmax-k problem for $k > 2$ drones by drawing connections to the multiway number partition problem and employing similar ideas as presented in Section 2.

Another avenue for future research is the enhancement of the G2D-algorithm. After getting $\{T_1, T_2\}$ (assume $\ell(T_2) \geq \ell(T_1)$), two tools can be employed for further improvement. With the *Cutting technique*, the idea is to break one tour of T_2 into two sub-tours and reducing the cost by allocating one sub-tour to the other drone as illustrated in Figure 4 (a).

On the other hand, the *Enlarging technique* can be applied when a tour $t_i \in T_1$ with a length strictly less than L is found to be in contact with a tour $t_j \in T_2$. Then we can simultaneously enlarge the tour t_i and reduce the length of the tour t_j (Figure 4 (b)). We plan to conduct a series of experiments to test how much the approximation factor improves with these tools.

References

1 Sergey Bereg, José-Miguel Díaz-Báñez, Alina Kasiuk, Miguel-Angel Pérez-Cutiño, and Fabio Rodríguez. Covering segments on a line with drones. In *Proc. 20th Spanish Meeting on Computational Geometry (EGC23)*, page 5, 2023.

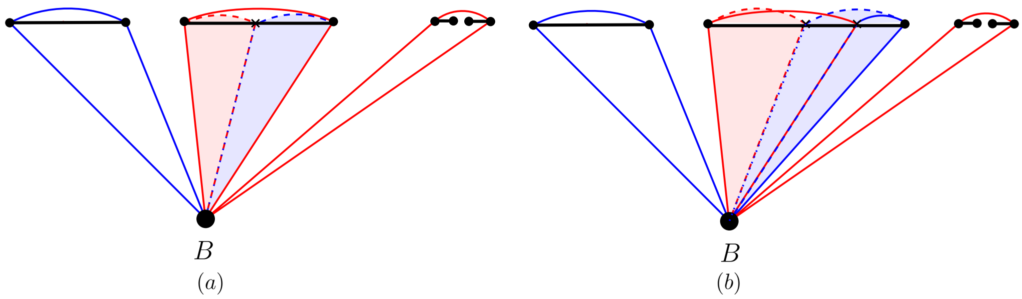

Figure 4 (a) Cutting a red tour. (b) Enlarging a blue tour and reducing a red tour.

2 James F. Campbell, Ángel Corberán, Isaac Plana, and José M. Sanchis. Drone arc routing problems. *Networks*, 72(4):543–559, 2018.

3 Michael R Garey. Computers and intractability: A guide to the theory of np-completeness, freeman. *Fundamental*, 1997.

4 Giusy Macrina, Luigi Di Puglia Pugliese, Francesca Guerriero, and Gilbert Laporte. Drone-aided routing: A literature review. *Transportation Research Part C: Emerging Technologies*, 120:102762, 2020.

5 Miguel-Angel Pérez-Cutiño, Juan Valverde, and José-Miguel Díaz-Báñez. Detecting broken receiver tubes in csp plants using intelligent sampling and dual loss. *Applied Intelligence*, pages 1–16, 2023.

6 Sékou Togola, Sountongnoma Martial Anicet Kiemde, and Ahmed Dooguy Kora. Real time and post-processing flight inspection by drone: A survey. In *Proc. 43rd International Conference on Telecommunications and Signal Processing (TSP)*, pages 399–402. IEEE, 2020.