Optimal Delivery with a Faulty Drone

Jared Coleman* Evangelos Kranakis† Danny Krizanc‡ Oscar Morales-Ponce§

Abstract

We introduce and study a new cooperative delivery problem inspired by drone-assisted package delivery. We consider a scenario where a drone moving in a straight line towards its destination loses communication (at time $t = 0$) with its central command station. The command station cannot know whether the drone's system has wholly malfunctioned or merely experienced a communications failure and the drone is continuing towards its destination. Consequently, a second helper drone with the same speed is deployed from the command station to retrieve the package and ensure successful delivery should the first drone fail before reaching its destination. The central question of this study is to find an algorithm for this second drone with optimal competitive ratio with respect to an omniscient drone that knows exactly if and when the first drone will fail. We show that the optimal algorithm depends on a surprisingly intricate relationship between the relative initial positions of the two drones and the destination.

1 Introduction

In recent years, drone-based package delivery has emerged as a promising application of unmanned aerial vehicle (UAV) technology. As these systems are increasingly integrated into supply chain infrastructures, it becomes imperative to design algorithms for their robust operation amidst unexpected complications. In this paper, we consider a complication arising from faulty communication and propose a solution for a cooperative delivery problem inspired by this scenario.

Consider a situation where a drone, en route to deliver a package to a given destination, unexpectedly loses communication with its central command station. This unexpected loss of contact leaves the command station uncertain of whether the drone has suffered a communications breakdown or complete system failure. Furthermore, even if the issue *is* only with the communications, the command station no longer has any way of knowing if/where the drone will fail on the rest of its way to the destination. In order to guarantee the package gets delivered, the command station must dispatch a second helper drone to retrieve the package and complete the delivery. Our goal is to design an online algorithm (one that cannot anticipate the true fail location of the drone) that, given the drone's last known location, determines the best trajectory for the second drone to find the package and complete the delivery in minimal time.

Formally, let us denote the last known location of the drone as the origin $S = (0, 0)$, the destination as the point $T = (1, 0)$, and the location of the command station as $P = (x, y)$, where $y \geq 0$ (all without loss of generality). The task is to identify an optimal trajectory for the second drone that minimizes the competitive ratio when compared to an optimal offline algorithm that knows the exact failure location $(t, 0)$ of the first drone in advance. We assume the first drone will fail at some time $0 \leq t \leq 1$ (if it does not fail the delivery time is optimal and the problem is uninteresting). Also, to simplify notation, we only consider the package to be delivered once the *second* drone and the package are co-located at the destination (i.e., if $t = 1$ and the first drone fails *at* the destination, the package is only delivered once the second drone reaches T). For an algorithm $\mathcal{A}(x, y)$, which defines the trajectory of the second drone, let $A(x, y, t)$ denote the delivery time of algorithm $\mathcal{A}(x, y)$ for failure time t. Our goal is to find an online algorithm (where t is unknown) with minimum competitive ratio with respect to the delivery time of an optimal offline algorithm, $\text{Opt}(x, y, t)$ (where t is known ahead of time). The competitive ratio for algorithm $\mathcal{A}(x, y)$ for a given fail time t can be written as

$$\text{CR}_{\mathcal{A}(x,y)}(t) = \frac{A(x, y, t)}{\text{Opt}(x, y, t)}.$$

Then the competitive ratio of $\mathcal{A}(x, y)$ is

$$\text{CR}_{\mathcal{A}(x,y)} = \sup_t \text{CR}_{\mathcal{A}(x,y)}(t).$$

To simplify notation we sometimes eliminate x and y when they are clear from context and write algorithm $\mathcal{A}(x, y)$ as $\mathcal{A}$ and its delivery time $A(x, y, t)$ as $A(t)$.

1.1 Model and Notation

In this section, we describe the model and notation used throughout the paper. We call the first drone, which is

*Loyola Marymount University, jared.coleman@lmu.edu
†Carleton University, kranakis@scs.carleton.ca
‡Wesleyan University, dkrizanc@wesleyan.edu
§California State University, Long Beach, Oscar.MoralesPonce@csulb.edu

initially carrying the package towards the destination, the *starter* and the second drone, which completes the delivery, the *finisher*. Without loss of generality, let $S = (0, 0)$ be the initial location of the starter (the point where it loses communication with the central command station) and $T = (1, 0)$ be the destination point on the plane. The starter drone begins at point S carrying the package and moves following a straight line directly towards point T. At some unknown time $t \leq 1$, the starter will fail at position $(t, 0)$.

The finisher drone starts at a point $P = (x, y)$ on the plane (i.e., at the command station). Without loss of generality, we assume that $y \geq 0$ (all results follow trivially by symmetry). We assume that both drones have a speed of 1 and always move at this speed. The finisher can start, stop, and change direction instantaneously. The drones can communicate with each other and exchange the package only when they are co-located (face-to-face communication). The package is considered to be delivered as soon as the finisher and the package are co-located at T.

Let $D(c, r)$ (resp., $\overline{D}(c, r)$) denote the open (resp., closed) disk with radius r and center $(c, 0)$. We use capital letters to denote points, $|PQ|$ to denote the Euclidean distance between points P and Q, and $\overline{PQ}$ to denote the line segment with endpoints P and Q. To simplify notation we sometimes eliminate x and y when they are clear from context, including when writing derived quantities such as d, t_1, and others that are functions of (x, y).

1.2 Related Work

An important aspect of our problem is that the starter agent experiences a failure. Many problems with cooperative mobile agent experiencing failures have been studied for a variety of basic problems in distributed computing and in various domains. In [11], the authors study the search problem on the line by n mobile agents where f of the agents are faulty. They present algorithms and their competitive ratios for different values of f. In [13] the authors study the evacuation problem on the disk by n agents, f of which may be faulty. Competitive algorithms for gathering have been proposed for n agents in a synchronous system with less than $(n - 1)/3$ failures [1]. Optimal algorithms and hardness results for the multi-agent patrolling problem with faulty agents were presented in [9]. Algorithms for flocking [16] and evacuation [10] have also been studied in the context of faulty agents.

The problem studied in this paper has both search and delivery components. Many search problems have been studied for different domains and under different models and assumptions (cf. the book [2]). The authors of [6] consider the delivery problem for messages on the line segment by multiple agents with different speeds and propose competitively optimal algorithms. The results of the previous study were also extended to the plane [5]. Joint search and delivery problems, however, have received much less attention in the literature. The problem has also been studied for the single- and two-agent case on the line [4]. On the systems research side, many solutions for drone-assisted package delivery have been proposed for different environments and under different assumptions (we refer the reader to the survey [14]). There are also numerous studies on drone reliability. E.g., in [15] the authors consider the reliability of drones in a delivery network so as to minimize expected loss of demand assuming drone failures follow an exponential distribution and study the impact of including reliability in drone scheduling.

Most related but different to our delivery problem are the papers [3] and [12] on package delivery. In the former, the authors investigate delivery of one or two packages of many autonomous mobile agents initially located on distinct nodes of a weighted graph. In the latter paper, they are concerned with delivering a package from a source node to a destination node in a graph using a set of drones and study the setting where the movements of each drone are restricted to a certain subgraph of the given graph. Note that both papers above address the delivery problem in a graph setting. To the best of our knowledge our paper is the first to address delivery of a package in the plane in the presence of a faulty drone.

1.3 Results

The main result we present is an optimal online algorithm that depends only on the starting position of the finisher (the central command station). Essentially, the algorithm executes one of three candidate algorithms depending on the finisher's starting position. This is depicted in Figure 1. If the finisher starts in the diagonally striped region, then the optimal algorithm is for the finisher to go to S (the origin) and then towards the destination until finding the failed starter with the object. If the finisher starts in the vertically striped region, the optimal algorithm is for the finisher to go first to T (the destination) and then toward S until finding the failed starter with the object. Finally, if the finisher starts in the horizontally striped region, then the optimal algorithm is for the starter to go to the point $M = \left(\left(x^2 + y^2\right)/(2x), 0\right)$ and then toward S until finding the failed starter with the object. We will show that the finisher is guaranteed to find the starter on the interval $[0, M]$. While the behavior of the finisher is simple, the separating curves that mark the boundaries between these regions are non-linear and quite surprising. Intricate details of these curves are depicted in Figures 4, 5, and 6 (Section 4).

The rest of the paper is organized as follows.

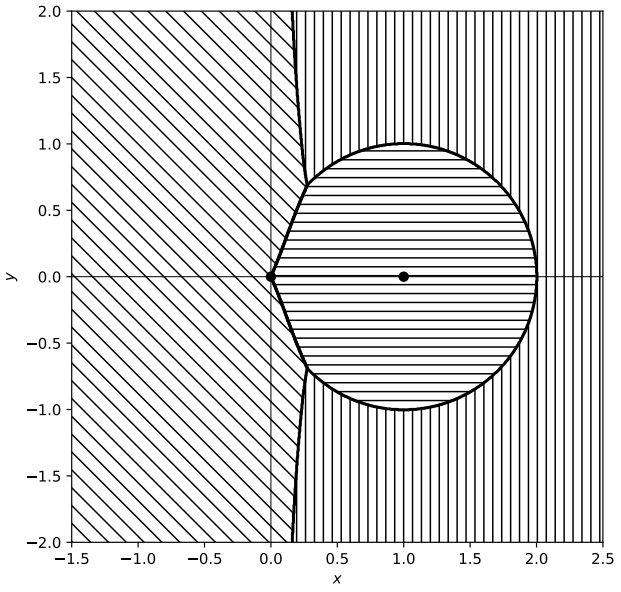

Figure 1: The optimal algorithm with $S = (0,0)$ and $T = (1,0)$ depends on the starting position of the finisher. The striped regions depict the finisher starting positions for which each of the three candidate algorithms is optimal.

Section 2 introduces three candidate algorithms and derives their competitive ratios as functions of the command station's starting position. Section 3 presents a hybrid algorithm that selects the best candidate based on the starting position (x, y); we prove this hybrid is optimal. Section 4 examines how the starting position influences the chosen candidate within the hybrid. Finally, Section 5 summarizes the results and outlines future directions. All proofs omitted due to space constraints can be found in the full version of the paper [8]. Some proofs rely on computations performed in Mathematica, which are publicly available on GitHub [7].

2 Candidate Algorithms

In this section, we present three algorithms and derive their competitive ratios. In order to do so, we must first consider the optimal offline algorithm, where the fail location $(t, 0)$ (at time t) is known ahead of time and can be used to compute the optimal trajectory for the finisher. Clearly in this case the finisher should go directly from its starting location to the starter's fail location $(t, 0)$ and then complete the delivery. The delivery time then can be written:

$$\mathrm{Opt}(t) = \max\left\{1, \sqrt{(x-t)^2 + y^2} + 1 - t\right\}.$$

Indeed, although the finisher moves directly to $(t, 0)$, it may have to wait for the starter to arrive (if necessary) prior to completing the delivery task.

For the online algorithms, we start by reasoning about what an optimal algorithm looks like. First, since the starter must fail at some point on the line segment $\overline{ST}$, the finisher must *eventually* move from its initial location to some point $(m, 0)$ on the segment (otherwise it will never find the starter with the package). We use this to prove the following intuitively obvious lemma:

Lemma 1 *There exists an online algorithm with optimal competitive ratio that involves the finisher moving from its initial position $P = (x, y)$ directly to a point $M = (m, 0) \in \overline{ST}$, past which, it remains within the line segment $\overline{ST}$.*

Now, we present three candidate online algorithms:

1. $\mathcal{A}_0$ (**Go To Last Point of Contact**): The finisher moves to the origin $S = (0, 0)$ and then towards the destination until it finds the failed starter (trajectory $P \to S \to T$).

2. $\mathcal{A}_1$ (**Go To Destination**): The finisher goes first to the destination $T = (1, 0)$ and then towards the origin until it finds the failed starter (trajectory $P \to T \to S \to T$).

3. $\mathcal{A}_d$ (**Meet in the Middle**): The finisher goes first to the point $(d, 0)$, where $d = (x^2 + y^2)/(2x)$, and then towards the origin until it finds the failed starter (trajectory $P \to (d, 0) \to S \to T$). The point $(d, 0)$ is the unique point on the line segment where the drones, both moving continuously, would meet simultaneously if the starter does not fail before time d.

Lemma 2 *$CR_{\mathcal{A}_d(x,y)} \le CR_{\mathcal{A}_1(x,y)}$ if and only if the finisher starts within distance 1 of the destination (i.e., $(x, y) \in \overline{D}(1, 1)$).*

Lemma 2 essentially tells us that we need only consider (among the candidate algorithms) $\mathcal{A}_0$ and $\mathcal{A}_d$ when the finisher starts *inside* the disk and algorithms $\mathcal{A}_0$ and $\mathcal{A}_1$ when the finisher starts *outside* the disk. Note that, when (x, y) is on the edge of the disk $\overline{D}(1, 1)$ (i.e., $(x-1)^2 + y^2 = 1$), then $d = 1$ and so $\mathcal{A}_d$ and $\mathcal{A}_1$ are the same algorithm.

2.1 Go to Last Point of Contact

In this section, we derive the competitive ratio of Algorithm $\mathcal{A}_0$. Observe that, no matter the failure time of the starter, the algorithm takes time $1 + \sqrt{x^2 + y^2}$ to deliver the message. This fact makes deriving the algorithm's competitive ratio rather simple.

Theorem 3 *The competitive ratio of $\mathcal{A}_0$ is*

$$CR_{\mathcal{A}_0} = \frac{1+\sqrt{x^2+y^2}}{\max\left\{1, \sqrt{(x-1)^2+y^2}\right\}}.$$

2.2 Go to Destination

We will now derive the competitive ratio of algorithm $\mathcal{A}_1$. Unlike the delivery time of algorithm $\mathcal{A}_0$, the delivery time $A_1(t) = \sqrt{(x-1)^2+y^2} + 2(1-t)$ (the time for the agent to go to $(1,0)$ and then backtrack until it finds the package at $(t,0)$ before returning to $(1,0)$ to complete the delivery) depends greatly on the fail time of the starter. Thus, to find the competitive ratio of algorithm $\mathcal{A}_1$, we must find the worst-case fail time for a given finisher starting position (x,y).

Theorem 4 *Assume $(x,y) \notin D(1,1)$ (i.e., $\sqrt{(x-1)^2+y^2} \geq 1$). The competitive ratio of $\mathcal{A}_1$ is $CR_{\mathcal{A}_1}(\max\{t_1,0\})$ where*

$$t_1 = \begin{cases} 1 - 3y/4 & x = 1 \\ \frac{x^2+y^2+z_1(1-x)-1-z_1\sqrt{x(x+z_1-2)+y^2-z_1+1}}{2(x-1)} & o.w. \end{cases}$$

and $z_1 = \sqrt{(x-1)^2+y^2}$.

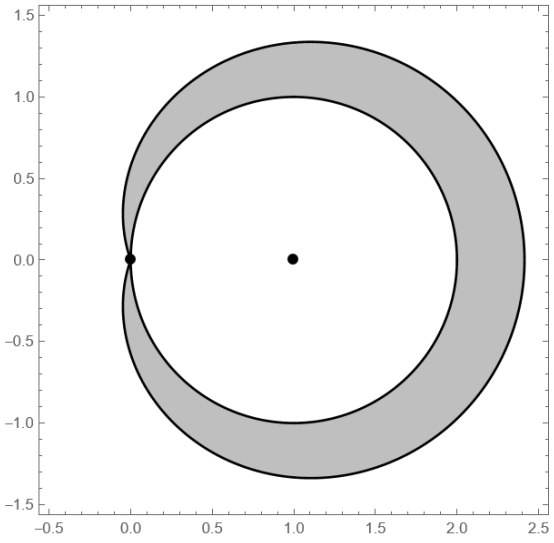

Figure 2: The worst-case fail time for the starter is 0 *except* in the gray shaded region (defined in the proof of Theorem 4, which can be found in the full version of the paper [8]). Recall that we only consider $\mathcal{A}_1$ when the finisher starts outside of the Disk $D(1,1)$.

There are essentially two cases that drive the competitive ratio of Algorithm $\mathcal{A}_1$: when t_1 (from the statement of Theorem 4) is less than or equal to 0 and when it is greater than 0. Observe the value of t_1 is determined by the finisher's starting position (x,y). When $t_1 \leq 0$ (i.e., when the finisher starts outside of the gray region shown in Figure 2), the competitive ratio is driven by the case where the starter fails at the origin (as soon as it loses contact with the command station). When $t_1 > 0$, though, the worst-case scenario for $\mathcal{A}_1$ is when the starter fails at time t_1.

2.3 Meet in the Middle

In this section, we derive the competitive ratio of the last candidate algorithm, $\mathcal{A}_d$, where the finisher moves from its starting position to the point $d = (x^2+y^2)/(2x)$, then towards the origin until it finds the starter with the package before completing the delivery. Recall from Lemma 2 that we need only consider algorithm $\mathcal{A}_d$ inside the closed disk centered at $(1,0)$ with radius 1, namely $\overline{D}(1,1)$. Furthermore, observe that on the edge of this disk $d = 1$ and so algorithms $\mathcal{A}_d$ and $\mathcal{A}_1$ are equal. By construction, then, the finisher *must* find the starter between the origin and the point $(d,0)$ since $(d,0)$ is the unique point on $\overline{ST}$ such that the distance between the origin and $(d,0)$ is equal to the distance from (x,y) to 0 (i.e., it is the point where the two drones would meet simultaneously if the starter does not fail). It is easy to derive that $d = (x^2+y^2)/(2x)$ by solving $\sqrt{(x-d)^2+y^2} = d$ for d using simple algebra. Thus, at time d, the finisher reaches point $(d,0)$ and the starter cannot have reached any point further than $(d,0)$.

Theorem 5 *Assume $(x,y) \in \overline{D}(1,1)$ (i.e., $\sqrt{(x-1)^2+y^2} \leq 1$). The competitive ratio of $\mathcal{A}_d$ is*

$$CR_{\mathcal{A}_d} = \begin{cases} \frac{x^2+y^2+x}{x\left(1+\sqrt{x^2+y^2}\right)} & \text{if } (x,y) \in \overline{D}(1/2,1/2) \\ 1 + \frac{y^2}{x\left(\sqrt{x}+1\right)^2} & \text{otherwise.} \end{cases}$$

Similar to Algorithm $\mathcal{A}_1$, two cases drive Algorithm $\mathcal{A}_d$'s competitive ratio. The first is when t' (from the proof of Theorem 5) is less than 0, whenever the starting position of the finisher is inside the disk $\overline{D}(1/2,1/2)$. In this case, the worst-case failure time for the starter is $t = 0$ (i.e., at the origin). When the finisher starts outside of the disk $\overline{D}(1/2,1/2)$ (but inside the disk $\overline{D}(1,1)$, of course, since we only consider Algorithm $\mathcal{A}_d$ in this region), however, the worst-case failure time for the starter is $t = t'$ (i.e., at location $(t',0)$, see Figure 3).

3 A Hybrid Algorithm

For convenience, we summarize our main result by introducing Algorithm 1, which simply executes the best of $\mathcal{A}_0$, $\mathcal{A}_1$, and $\mathcal{A}_d$ given a finisher starting position (x,y), and prove it to be optimal.

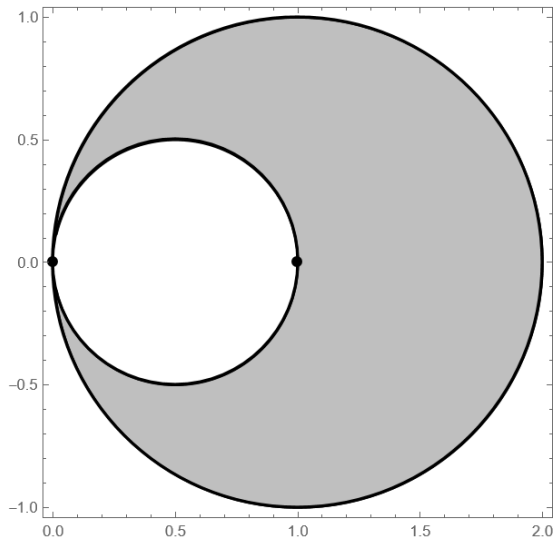

Figure 3: The worst-case fail time for the starter is 0 *except* in the gray shaded region, where it is t' (from the proof of Theorem 5). Recall that we only consider $\mathcal{A}_d$ when the finisher starts inside the Disk $\overline{D}(1,1)$.

Algorithm 1 A hybrid algorithm

```
1: input: Finisher starting position (x, y)
2: if (x − 1)^2 + y^2 > 1 then
3:     if CR_{A_0(x,y)} ≤ CR_{A_1(x,y)} then
4:         Execute Algorithm A_0
5:     else Execute Algorithm A_1
6: else
7:     if CR_{A_0(x,y)} ≤ CR_{A_d(x,y)} then
8:         Execute Algorithm A_0
9:     else Execute Algorithm A_d
```

In fact, we will show that Algorithm 1 is optimal by proving that any other algorithm A_a that moves first to a position $(a, 0)$ such that $0 < a < 1$ and $a \neq d$ is always worse than at least one of the candidate algorithms $\mathcal{A}_0$, $\mathcal{A}_1$, or $\mathcal{A}_d$. Recall from Lemma 1 that we do not need to consider any other algorithms. Leveraging Lemma 2, we know that we only need consider $\mathcal{A}_0$ and $\mathcal{A}_d$ when the finisher's starting position (x, y) is inside the disk $\overline{D}(1,1)$ and algorithms $\mathcal{A}_0$ and $\mathcal{A}_1$ when it is outside the disk. The following lemmas cover each of these cases.

Lemma 6 *For any* $(x, y) \in \overline{D}(1,1)$, $\min\{CR_{\mathcal{A}_d}, CR_{\mathcal{A}_0}\} \leq CR_{\mathcal{A}_a}$ *for all* $a \in \overline{ST}$.

Again, by Lemma 2, when the finisher starts outside of the disk $D(1,1)$, we need only consider $\mathcal{A}_0$ and $\mathcal{A}_1$.

Lemma 7 *For any* $(x, y) \notin D(1,1)$, *either* $CR_{\mathcal{A}_1} \leq CR_{\mathcal{A}_a}$ *or* $CR_{\mathcal{A}_0} \leq CR_{\mathcal{A}_a}$ *for all* $a \in \overline{ST}$.

Theorem 8 *Algorithm 1 is optimal.*

Proof. Follows directly from Lemmas 6 and 7. □

4 Discussion

We now examine in more detail the regions where each candidate algorithm is optimal, as shown in Figures 4, 5, and 6. While the plots were generated numerically, we are able to obtain a closed-form equation for one of the separating curves.

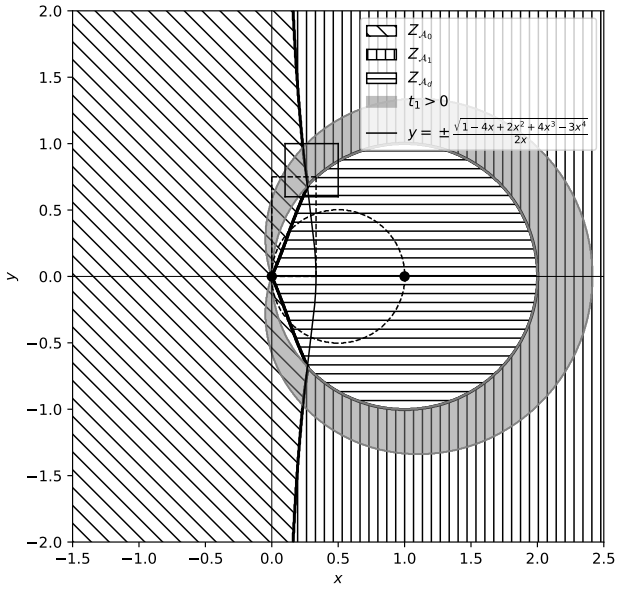

Figure 4: Plot showing which algorithm has the least competitive ratio given finisher starting position (x, y).

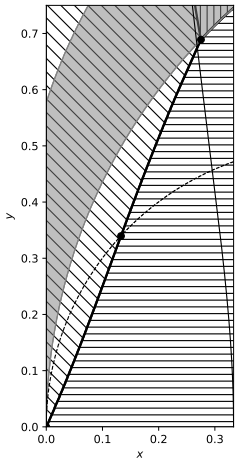

Figure 5: Region in Figure 4 bounded by dashed rectangular box.

The competitive ratio of each of the candidate algorithms, and therefore the hybrid algorithm, depends on the starting position of the finisher. Let $Z_{\mathcal{A}_0}$ denote the set of points (x, y) such that Algorithm 1 executes

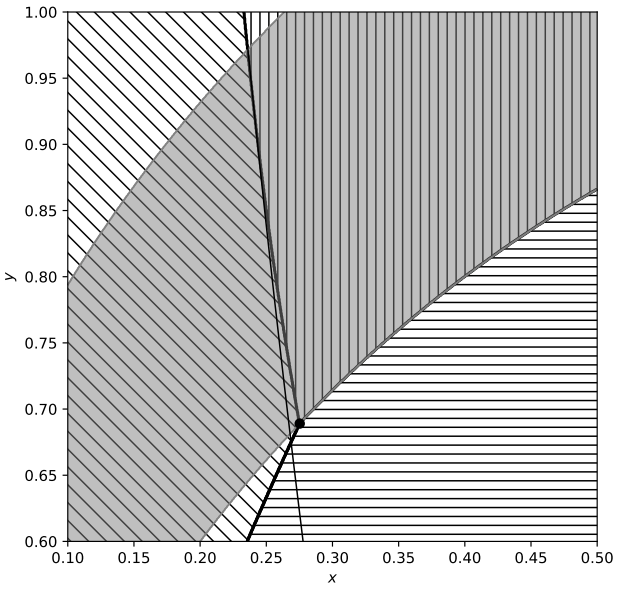

Figure 6: Region in Figure 4 bounded by solid-bordered square box.

$\mathcal{A}_0$ (i.e., $(x-1)^2+y^2 > 1$ and $\mathrm{CR}_{\mathcal{A}_0(x,y)} \leq \mathrm{CR}_{\mathcal{A}_1(x,y)}$, or else $(x-1)^2+y^2 \leq 1$ and $\mathrm{CR}_{\mathcal{A}_0(x,y)} \leq \mathrm{CR}_{\mathcal{A}_d(x,y)}$). Similarly, let $Z_{\mathcal{A}_1}$ denote the set of points (x,y) such that Algorithm 1 executes $\mathcal{A}_1$ (i.e., $(x-1)^2+y^2 > 1$ and $\mathrm{CR}_{\mathcal{A}_0(x,y)} > \mathrm{CR}_{\mathcal{A}_1(x,y)}$) and $Z_{\mathcal{A}_1}$ the set of points (x,y) such that Algorithm 1 executes $\mathcal{A}_d$ (i.e., $(x-1)^2+y^2 \leq 1$ and $\mathrm{CR}_{\mathcal{A}_0(x,y)} > \mathrm{CR}_{\mathcal{A}_d(x,y)}$). In other words, for a finisher starting position (x,y), the hybrid algorithm executes algorithm $\mathcal{A}_0$ iff $(x,y) \in Z_{\mathcal{A}_0}$, $\mathcal{A}_1$ iff $(x,y) \in Z_{\mathcal{A}_1}$, and $\mathcal{A}_d$ iff $(x,y) \in Z_{\mathcal{A}_d}$. Observe the regions $Z_{\mathcal{A}_0}$, $Z_{\mathcal{A}_1}$, and $Z_{\mathcal{A}_d}$ are disjoint and their union comprises the entire plane. Figure 4 depicts these regions.

Observe if (x,y) is inside the disk $D(1,1)$, either $\mathcal{A}_0(x,y)$ or $\mathcal{A}_d(x,y)$ has the least competitive ratio. Otherwise, if $(x,y) \notin \overline{D}(1,1)$ either $\mathcal{A}_0(x,y)$ or $\mathcal{A}_1(x,y)$ has the least competitive ratio. This is consistent, of course, with Lemma 2. Inside each of these regions, there are even more interesting things going on. Examine the bang-bang curve (i.e., the separating curve) between the regions $Z_{\mathcal{A}_0}$ and $Z_{\mathcal{A}_d}$ inside the disk $D(1/2,1/2)$. Recall that the worst-case fail time for $\mathcal{A}_d(x,y)$ is $t_d = \frac{x(x-1)+y^2}{2(x+\sqrt{x})}$ if $(x,y) \in D(1/2,1/2)$ and 0 otherwise. This is what causes the change of inflection when the bang-bang curve leaves the disk $D(1/2,1/2)$.

Now, look at the bang-bang curve between $Z_{\mathcal{A}_0}$ and $Z_{\mathcal{A}_1}$ outside of the disk $D(1,1)$. Recall the worst-case fail time for $\mathcal{A}_1(x,y)$ is

$$t_1 = \begin{cases} 1 - 3y/4 & \text{if } x = 1 \\ \frac{x^2+y^2+z_1(1-x)-1-z_1\sqrt{b}}{2(x-1)} & \text{otherwise} \end{cases}$$

(where $z_1 = \sqrt{(x-1)^2+y^2}$ abd $b = x(x+z_1-2) + y^2 - z_1 + 1$) if $t_1 > 0$ and 0 otherwise. Outside of this region, we can find a closed-form equation for the curve separating $\mathcal{A}_0$ and $\mathcal{A}_1$ since the competitive ratios for $\mathrm{CR}_{\mathcal{A}_0}$ and $\mathrm{CR}_{\mathcal{A}_1}$ are relatively simple. We can find where the competitive ratios are equal for this case:

$$\mathrm{CR}_{\mathcal{A}_1(x,y)}(0) = \mathrm{CR}_{\mathcal{A}_0(x,y)}$$
$$\frac{\sqrt{(x-1)^2+y^2}+2}{\sqrt{x^2+y^2}+1} = \frac{\sqrt{x^2+y^2}+1}{\sqrt{(x-1)^2+y^2}}.$$

By cross-multiplying the right-hand side and simplifying, we obtain:

$$-2x+1 = 2x\sqrt{x^2+y^2} + x^2$$

From this, it is easy to arrive at:

$$y = \pm\sqrt{\left(\frac{1-2x-x^2}{2x}\right)^2 - x^2}$$
$$= \pm\frac{\sqrt{1-4x+2x^2+4x^3-3x^4}}{2x} \qquad (4.1)$$

Unfortunately, this is the only bang-bang curve for which we were able to find a closed form equation. Observe in Figure 6 how the bang-bang curve diverges from the curve given by Equation (4.1) when $t_1 > 0$ (inside the shaded region).

5 Conclusion

In this paper, we present a competitively optimal algorithm for two-agent delivery in the plane with a single faulty agent. We show that the competitive ratio of the algorithm depends on the relative positioning of the two agents and the destination. An upper bound of 3 on the competitive ratio over all points (x,y) is straightforward to prove (both $\mathrm{CR}_{\mathcal{A}_0(x,y)}$ and $\mathrm{CR}_{\mathcal{A}_1(x,y)}$ have a maximum of 3). Numerical calculations indicate the maximum competitive ratio is given by the case where $\mathrm{CR}_{\mathcal{A}_0(x,y)} = \mathrm{CR}_{\mathcal{A}_1(x,y)} = \mathrm{CR}_{\mathcal{A}_d(x,y)}$, which is approximately 1.74197 at $x \approx 0.275257, y \approx 0.689019$. The results presented in this paper introduce a number of interesting questions and avenues for future work. First, we assume the starter can only move directly toward the destination. It would be interesting to see how the competitive ratio changes if the starter can move in any direction (e.g., toward the finisher). Second, we could remove the assumption that the starter/finisher move at the same speed. In this case, the finisher might be able to deliver the package faster by participating even if the starter does not fail. We could also extend the problem to consider multiple (potentially faulty) agents (i.e., what happens if the finisher also fails?). Finally, another interesting open problem arises when the starter's trajectory from the source to the target is on a curve other than a line segment.

References

[1] N. Agmon and D. Peleg. Fault-tolerant gathering algorithms for autonomous mobile robots. *SIAM J. Comput.*, 36(1):56–82, 2006.

[2] S. Alpern and S. Gal. *The theory of search games and rendezvous*, volume 55 of *International series in operations research and management science*. Kluwer, 2003.

[3] I. A. Carvalho, T. Erlebach, and K. Papadopoulos. On the fast delivery problem with one or two packages. *J. Comput. Syst. Sci.*, 115:246–263, 2021.

[4] J. Coleman, L. Cheng, and B. Krishnamachari. Search and rescue on the line. In *Structural Information and Communication Complexity*, pages 297–316, Cham, 2023. Springer Nature Switzerland.

[5] J. Coleman, E. Kranakis, D. Krizanc, and O. Morales-Ponce. Message delivery in the plane by robots with different speeds. In *Stabilization, Safety, and Security of Distributed Systems - 23rd International Symposium, SSS 2021, Virtual Event, November 17-20, 2021, Proceedings*, volume 13046 of *Lecture Notes in Computer Science*, pages 305–319. Springer, 2021.

[6] J. Coleman, E. Kranakis, D. Krizanc, and O. Morales-Ponce. The pony express communication problem. In *Combinatorial Algorithms - 32nd International Workshop, IWOCA 2021, Ottawa, ON, Canada, July 5-7, 2021, Proceedings*, volume 12757 of *Lecture Notes in Computer Science*, pages 208–222. Springer, 2021.

[7] J. Coleman, D. Krizanc, E. Kranakis, and O. Morales-Ponce. Mathematica code for faulty delivery problem. https://github.com/jaredraycoleman/faulty_delivery, 2025. Accessed: 2025-06-20.

[8] J. R. Coleman, D. Krizanc, E. Kranakis, and O. Morales-Ponce. Optimal delivery with a faulty drone. *CoRR*, abs/2404.17711, 2024.

[9] J. Czyzowicz, L. Gasieniec, A. Kosowski, E. Kranakis, D. Krizanc, and N. Taleb. When patrolmen become corrupted: Monitoring a graph using faulty mobile robots. *Algorithmica*, 79(3):925–940, 2017.

[10] J. Czyzowicz, K. Georgiou, M. Godon, E. Kranakis, D. Krizanc, W. Rytter, and M. Wlodarczyk. Evacuation from a disc in the presence of a faulty robot. In *Structural Information and Communication Complexity - 24th International Colloquium, SIROCCO 2017, Porquerolles, France, June 19-22, 2017, Revised Selected Papers*, volume 10641 of *Lecture Notes in Computer Science*, pages 158–173. Springer, 2017.

[11] J. Czyzowicz, E. Kranakis, D. Krizanc, L. Narayanan, and J. Opatrny. Search on a line with faulty robots. *Distributed Comput.*, 32(6):493–504, 2019.

[12] T. Erlebach, K. Luo, and F. C. R. Spieksma. Package delivery using drones with restricted movement areas. In *33rd International Symposium on Algorithms and Computation, ISAAC 2022, December 19-21, 2022, Seoul, Korea*, volume 248 of *LIPIcs*, pages 49:1–49:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[13] K. Georgiou, E. Kranakis, N. Leonardos, A. Pagourtzis, and I. Papaioannou. Optimal circle search despite the presence of faulty robots. In *Algosensors 2019, Munich, Germany, September 12-13, 2019*, volume 11931 of *LNCS*, pages 192–205. Springer, 2019.

[14] A. M. Raivi, S. M. A. Huda, M. M. Alam, and S. Moh. Drone routing for drone-based delivery systems: A review of trajectory planning, charging, and security. *Sensors*, 23(3):1463, 2023.

[15] M. Torabbeigi, G. J. Lim, and S. J. Kim. Drone delivery schedule optimization considering the reliability of drones. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1048–1053. IEEE, 2018.

[16] Y. Yang, S. Souissi, X. Défago, and M. Takizawa. Fault-tolerant flocking for a group of autonomous mobile robots. *J. Syst. Softw.*, 84(1):29–36, 2011.