Beam It Up, Scotty: Angular Freeze-Tag with Directional Antennas*

Sándor P. Fekete[1] and Dominik Krupke[1]

1 **TU Braunschweig**
{s.fekete,d.krupke}@tu-bs.de

Abstract

We consider distributing mission data among the members of a satellite swarm. In this process, spacecraft cannot be reached all at once by a single broadcast, because transmission requires the use of highly focused directional antennas. As a consequence, a spacecraft can transmit data to another satellite only if its antenna is aiming right at the recipient; this may require adjusting the orientation of the transmitter, incurring a time cost proportional to the required angle of rotation. The task is to minimize the total distribution time. This makes the problem similar in nature to the *Freeze-Tag Problem* of waking up a set of sleeping robots, but with angular cost at vertices, instead of distance cost along the edges of a graph. We prove that approximating the minimum length of a schedule for this *Angular Free-Tag Problem* within a factor of less than 5/3 is NP-complete, and provide a 9-approximation for the 2-dimensional case that works even in online settings with incomplete information. Furthermore, we develop an exact method based on Mixed Integer Programming that works in arbitrary dimensions and can compute provably optimal solutions for benchmark instances with about a dozen satellites.

1 Introduction

Providing instructions to all members of a distributed group is a fundamental task for many types of team missions. In terrestrial settings, this can usually be achieved by broadcasting to all recipients in parallel, requiring only a single transmission. However, for long-distance space missions, omnidirectional transmission can no longer be employed, due to significant loss in signal strength. Instead, transferring data is accomplished with the help of directional antennas, requiring a highly focused communication beam that is targeted right at the intended recipient. (See Figure 1 for an illustration.) As a consequence, these transmissions must be performed individually, involving maneuvers for achieving appropriate antenna orientation; the time for such a maneuver is basically proportional to the required angle of rotation, with negligible time for the actual transmission itself. The overall process does allow one parallel component: a team member that has already been "activated" by having received the data may relay this to other partners, motivating the use of intricate communication trees for achieving rapid dissemination of information to all members of a swarm of spacecraft.

This can be utilized if we want to quickly distribute data, e.g., an important update. In the following we consider a basic version of the problem in which the agents are static points in the euclidean space, there are no delays for transmission, and the transmission cone is modeled as a ray. (Also note that more advanced scenarios for space missions may require *both* a transmitting and a receiving antenna that are directed at the communication partner; see the Conclusions in Section 5.)

* Partially supported by the European Space Agency, project ASIMOV, contract number 4000122514/17/F/MOS.

34th European Workshop on Computational Geometry, Berlin, Germany, March 21–23, 2018.
This is an extended abstract of a presentation given at EuroCG'18. 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.

▶ Problem 1.1. **Angular Freeze Tag (AFT).** Given a set $P = \{p_0, \ldots, p_n\}$ of agent positions in d-dimensional space. Each agent $p_i \in P$ has an initial heading α_i. At time $t = 0$, only p_0 is *active*, while all other agents are *inactive*. An agent p_i is *activated* by an active agent p_j whose heading α_j aims right at p_i; adjusting this heading incurs a cost equal to the required angular change. The objective is to minimize the time T until all agents are activated, i.e., minimize the makespan of the overall activation schedule.

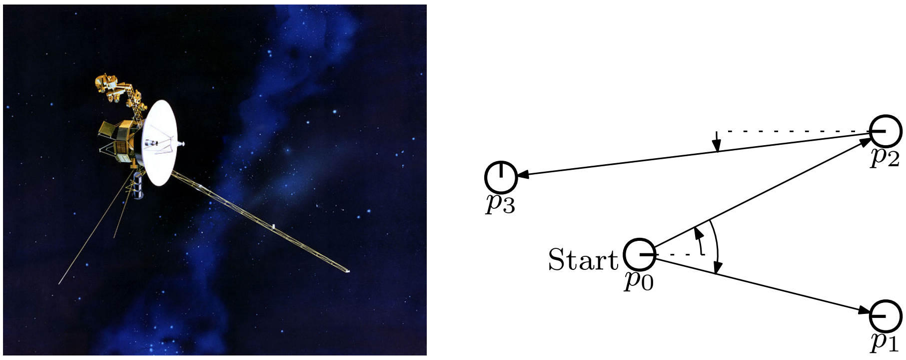

Figure 1 (Left) The space probe Voyager and its directional antenna for transmitting data. (Image CC by NASA.) (Right) Activating all agents by rotations: p_0 first activates p_2 which then activates p_3 while p_0 rotates back to activate p_1.

Related Work. The original *Freeze-Tag Problem* (FTP) was introduced by Arkin et al. [2], who studied the task of waking up a swarm of robots. In the FTP, activating an inactive robot is performed by moving an active robot next to it. The objective (minimize the makespan of the overall schedule) is the same as for our problem, but the cost for an activation (the distance to the robot instead of the angle) is different. This problem is NP-hard even for star graphs, but there are polynomial-time approximation schemes (PTAS) for star graphs and geometrically embedded instances [3]. Unweighted graphs are considered in [4]. A set of heuristics is evaluated in [11]. Results on the hardness in Euclidean space are provided by [1] and [9].

Other geometric questions related to the use of directional antennas have also been considered. Carmi et al. [8] studied the α-MST, which arose from finding orientations of directional antennas with α-cones, such that the connectivity graph yields an MST of minimum weight, based on bidirectional communication. They prove that for $\alpha < \pi/3$, a solution may not exist, while $\alpha \geq \pi/3$ always suffices. See Aschner and Katz [5] for more recent hardness proofs and constant-factor approximations for some α.

2 Hardness of Approximation

We show that the AFT is computationally hard, even to approximate.

▶ **Theorem 2.1.** *A c-approximation algorithm for the AFT with $c < 5/3$ implies $P = NP$.*

Proof. We give a reduction from Satisfiability; see Figure 2 for a sketch. Our construction has a solution with a makespan of 3ε if it is satisfiable and 5ε otherwise, where $\varepsilon > 0$ is a sufficiently small angle. Our construction uses five different types of agents, as follows.

- The *start agent* p_0 directly activates the *decision agents*, but does not have any other agents within 5ε of α_i.

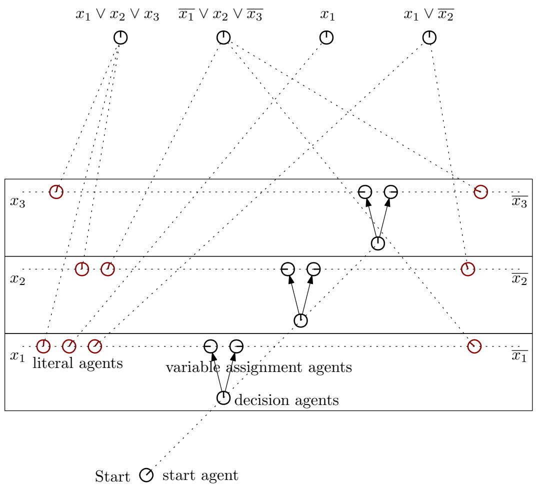

Figure 2 Sketch of the hardness construction. Red variable agents are 2ε from their designated heading, which they can target upon activation. The decision agent for each variable is a rotation ε from both of its corresponding variable assignment agents. A schedule of makespan 3ε exists if and only if there is a satisfying truth assignment; otherwise, the makespan is at least 5ε.

- For each variable we have a *decision agent* and two *variable assignment agents* (one each for `true` and `false`) in opposing angles of ε, but no further agents within a 5ε rotational range. It is directly activated from p_0.
- The *variable assignment agent* directly activates all corresponding *literal agents*, but has no further agents in a 4ε rotation range. The earliest possible activation time is ε. Only one of the two agents can be activated at time ε (by the *decision agent*), the other one has to wait an additional 2ε.
- For each literal there is a *literal agent* that has its clause agent a rotation of 2ε away, but no further agents within 4ε. The earliest possible activation time is ε.
- For each clause there is a *clause agent* that has no agent within its 2ε rotation range. Its earliest possible activation time is 3ε.

A clause agent can only be activated by its literal agents in less than 5ε and a literal agent is either activated at ε or 3ε, depending on which of the variable assignment agents got activated first. Thus, a clause is activated at 3ε if and only if a corresponding variable agent has been activated in time; otherwise, it takes 5ε. ◀

3 Approximation Algorithm

We can provide a simple constant factor approximation, based on a result by Beck [6] on the *linear search problem*. In that scenario, an agent has to locate a hidden object in a one-dimensional environment; from a given starting location, the best strategy for this online problem is to alternate between going left and right, while doubling the search depth in each iteration. This yields a total search distance that is within a factor of 9 of the optimum.

▶ **Theorem 3.1.** *There is a 9-approximation algorithm for the AFT in 2-dimensional space, even for unknown agent locations and headings, assuming a lower bound of $\varepsilon > 0$ for the rotational angle of any activating agent.*

Proof. As soon as an agent is activated, it follows the doubling strategy from linear search, carried out for rotation. It follows straightforward by induction that any agent p_i that gets

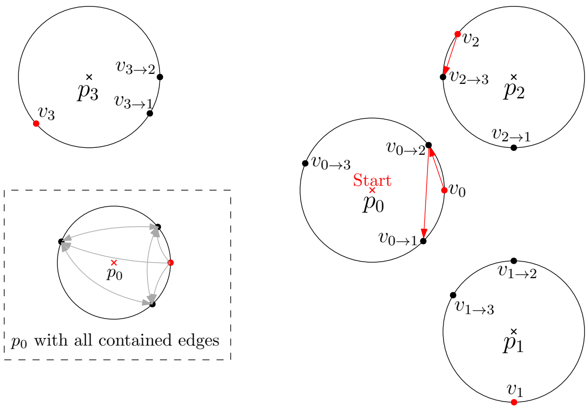

Figure 3 An example of the auxiliary graph. Every point (p_0, p_1, p_2, p_3) has a vertex for its initial heading (v_i) and a vertex for the heading to any other point different from p_0. Between the vertices of the same point, there are directed edges with the cost of the corresponding rotation; as shown in the lower left, there are no incoming edges for the start vertex. If points are collinear, there can be two vertices for the same heading. A possible solution would be for p_0 to first head to p_2 and then to p_1, while p_2 heads to p_3. The corresponding movements are visualized by red edges.

activated by T_i in an optimal schedule is activated within $9T_i$. ◀

Note that we cannot apply the refined technique by Bose et al. [7] for linear search, as it requires both an upper and a lower bound on the search distance.

4 Exact Solution

In the following, we describe the set of solutions by a Mixed Integer Program (MIP). This allows us to use an advanced solver such as CPLEX to obtain provably optimal solutions.

Each agent has only a finite set of relevant headings; between such two configurations there is an easily computable optimal rotation. The relevant configurations are the initial heading of an agent and the headings that activate other agents, for a total of $O(|P|)$ configurations and $O(|P|^2)$ transitions per agent. We can encode this into an auxiliary directed graph $G = (V, E)$ in which the configurations are the vertices and the vertices of each agent form a weakly connected component. For an agent $p_i \in P$ we denote the initial heading vertex by v_i, and the vertices that activate another agent $p_j \in P$ by $v_{i\to j}$. There is a directed edge between all vertices $v_{i\to j}, p_j \in P \setminus \{p_i, p_0\}$ as well as from v_i to all $v_{i\to j}, p_j \in P \setminus \{p_i, p_0\}$. There are no edges between the vertices of different agents. The movement (and agent activations) of an agent p_i can be represented by a directed path starting at v_i. Figure 3 visualizes such a graph and how to encode a solution.

We use *Boolean variables* $x_e, e \in E$ that represent the transition of an agent between two configurations, and *continuous variables* $y_v, v \in V$ that represent the time at which an agent reaches a specific configuration. If the configuration is not used, it may be zero. The value needs only to be tight for configurations that are critical for the makespan.

The general idea of the Mixed Integer Program is simple: the usage of an edge implies that the target's time has to be the source's time plus the transition time; we want to minimize the maximum value. It is also fairly simple to adapt this MIP to other problem variants. Let us start with the objective function that minimizes the latest activation time

$$\min \max_{p_i \in P} y_{v_i}. \tag{1}$$

Note that we need to implement the min-max via $\min t$ and $t \geq y_{v_i} \forall p_i \in P$, resulting in $O(|P|)$ additional constraints. For every agent we need to visit a vertex that activates it, i.e., we need to use an edge that visits such a vertex (exactly one to be precise).

$$\sum_{e \in E_{\text{in}}(v_{j\to i}), p_j \in P} x_e = 1 \quad \forall p_i \in P \setminus \{p_0\}. \tag{2}$$

Next we enforce that there are only directed paths starting at initial heading vertices by enforcing that there can only be at most one outgoing edge per vertex and only if there is also an ingoing one (3) or it is a start vertex (4), and prohibiting subcycles (5).

$$\sum_{E_{\text{out}}(v_{i\to j})} x_e \leq \sum_{E_{\text{in}}(v_{i\to j})} x_e \leq 1 \qquad \forall v_{i\to j} \in V \tag{3}$$

$$\sum_{E_{\text{out}}(v_i)} x_e \leq 1 \qquad \forall p_i \in P \tag{4}$$

$$\sum_{v,w \in S} x_{vw} \leq |S| - 1 \qquad \forall S \subset V \tag{5}$$

If agent p_i is activated by agent p_j, then $y_{v_i} = y_{v_{j\to i}}$. Since $y_{v_{k\to i}} = 0$ for all other agents p_k, we can write

$$y_{v_i} = \sum_{p_j \in P} y_{v_{j\to i}} \quad \forall p_i \in P \setminus \{p_0\}. \tag{6}$$

If we use a directed edge, we know that the target has to have the time of the source plus the minimal transition time, i.e., for an edge $vw \in E : y_w \geq y_v + \text{cost}(vw)$. We can neutralize this constraint by adding a large negative value to the right side that lowers it below zero if the edge is not selected. This value only needs to be 3π, because no optimal solution is larger than 2π and an edge cost is at most π.

$$y_w \geq y_v + \text{cost}(vw) + (3\pi x_{vw} - 3\pi) \quad \forall vw \in E \tag{7}$$

This constraint also prevents cyclic activations or cycles as in constraint (5) as long as they are not based on zero-cost transitions (this works analogous to the Miller-Tucker-Zemlin subtour elimination constraints for TSP [10]). To also prevent zero-cost cyclic activations we can use the following constraint:

$$\sum_{p_i, p_j \in S} \sum_{e \in E_{\text{in}}(v_{i\to j})} x_e \leq |S| - 1 \quad \forall S \subset P \setminus \{p_0\}. \tag{8}$$

Because this only happens for degenerated cases with zero-cost edges, we add the constraints (5) and (8) iteratively only if necessary.

In the end we have $\Theta(|P|^2)$ continuous variables, $\Theta(|P|^3)$ Boolean variables (of which only $|P|-1$ variables will be `true`), and $\Theta(|P|^3)$ constraints (excluding (5) and (8)), resulting in a relatively large problem that also becomes very quickly hard to solve, as can be seen in Fig. 4. Interestingly, this is not because CPLEX does not find a solution, but because it does not find an effective lower bound. Code on `https://github.com/d-krupke/eurocg18-angularft`.

5 Conclusion

We provided first results for a basic version of Angular Freeze Tag. Even in 2D with static transmitters, we need better lower bounds to improve approximation and the size of optimally solvable instances. There is also a wide spectrum of practically important generalizations.

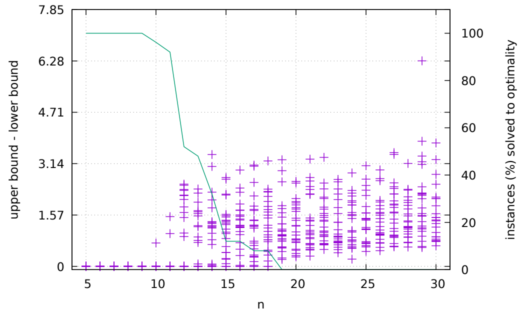

Figure 4 Results for random instances with CPLEX and a 15 min time limit on a PC (i7, 64GB). For 12 points only 50% can be solved to optimality. For unsolved instances, the lower bound is often close to zero, so providing better lower bounds will drastically improve performance.

These include approximation for the three-dimensional version and scenarios with moving satellites. Allowing inactive receivers to adjust their heading ahead of time may greatly speed up schedules. On the other hand, advanced missions may require *both* partners in a data exchange to have their directional antennas pointing at each other, making the scheduling process considerably more involved. All these issues are left for future work.

References

1. Z. Abel, H. A. Akitaya, and J. Yu. Freeze tag awakening in 2D is NP-hard. In *Proc. 27th Fall Worksh. Comp. Geom. (FWCG)*, 2017. Paper 28.
2. E. M. Arkin, M. A. Bender, S. P. Fekete, J. S. Mitchell, and M. Skutella. The freeze-tag problem: How to wake up a swarm of robots. In *Proc. 13th Symp. Disc. Alg. (SODA)*, pages 568–577, 2002.
3. E. M. Arkin, M. A. Bender, S. P. Fekete, J. S. Mitchell, and M. Skutella. The freeze-tag problem: how to wake up a swarm of robots. *Algorithmica*, 46(2):193–221, 2006.
4. E. M. Arkin, M. A. Bender, and D. Ge. Improved approximation algorithms for the freeze-tag problem. In *Proc. 15th Symp. Par. Alg. Arch. (SPAA)*, pages 295–303, 2003.
5. R. Aschner and M. J. Katz. Bounded-angle spanning tree: modeling networks with angular constraints. *Algorithmica*, 77(2):349–373, 2017.
6. A. Beck and D. J. Newman. Yet more on the linear search problem. *Israel Journal of Mathematics*, 8(4):419–429, 1970.
7. P. Bose, J. D. Carufel, and S. Durocher. Searching on a line: A complete characterization of the optimal solution. *Theoretical Computer Science*, 569:24–42, 2015.
8. P. Carmi, M. J. Katz, Z. Lotker, and A. Rosén. Connectivity guarantees for wireless networks with directional antennas. *Computational Geometry*, 44(9):477–485, 2011.
9. M. P. Johnson. Easier hardness for 3D freeze-tag. In *Proc. 27th Fall Worksh. Comp. Geom. (FWCG)*, 2017. Paper 29.
10. C. E. Miller, A. W. Tucker, and R. A. Zemlin. Integer programming formulation of traveling salesman problems. *J. ACM*, 7(4):326–329, 1960.
11. M. O. Sztainberg, E. M. Arkin, M. A. Bender, and J. S. Mitchell. Analysis of heuristics for the freeze-tag problem. In *Proc. 8th Scand. Worksh. Alg. Theo. (SWAT)*, pages 270–279, 2002.