The angular blowing-a-kiss problem

Kevin Buchin, Irina Kostitsyna, Roel Lambers, and Martijn Struijs

Department of Mathematics and Computing Science, TU Eindhoven, The Netherlands
k.a.buchin@tue.nl, i.kostitsyna@tue.nl, r.lambers@tue.nl, m.a.c.struijs@tue.nl

Abstract

Given a set of agents that have fixed locations but can rotate at unit speed, we aim to find an efficient schedule such that every pair of agents has looked at each other. We present schedules and lower bounds for different geometric settings.

1 Introduction

Given n people in a rectangular room, the *kissing problem* asks for the most efficient way for each pair of people to kiss each other goodbye [1]. We consider the variant of the problem in which the people blow kisses instead. Rather than changing locations, people now only need to turn to face each other.

Our motivation to study this problem comes from the research performed at NASA ARC on probing the magnetosphere of the Earth by a swarm of satellites with the use of directional antennas [2]. To perform probing, two satellites need to orient the antennas towards each other to be able to send and receive data. In this context we are interested in the most efficient schedule that allows every pair of satellites to perform probing. Independently, Fekete et al. [4] have studied this setting, focusing on the case in which only a subset of the satellite pairs need to communicate.

Problem Statement. In this paper, an *agent* has a fixed location in the plane and a heading direction that it can change over time at unit speed. We will refer to an agent by its location.

The input is a set of n agent locations $p_1, \ldots, p_n$ in the plane. Note that we allow agents to choose their initial direction. A pair of agents p_i, p_j can *scan* each other at time t if p_j is in the direction of p_i, and vice versa. We also say, the pair of agents is scanned. The goal is to define valid schedules for the agents as to minimize the time to scan all pairs of agents.

We define a schedule for all agents by defining a schedule per agent. A schedule for agent p_i is an ordering of the other agents $\Pi^i = \langle \pi^i_1, \ldots, \pi^i_{n-1} \rangle$ with a time t^i_j associated with each of the agents in the order. A schedule for the agents is valid if

1. $t^i_j \leq t^i_{j+1}$, for all i and for $0 \leq j < n$,
2. $\measuredangle \pi^i_j, p_i, \pi^i_{j+1} \leq t^i_{j+1} - t^i_j$, for all i and for $0 \leq j < n$, where $\measuredangle BAC$ denotes the smaller angle between B and C at A,
3. if $p_j = \pi^i_k$ and $p_i = \pi^j_\ell$, then $t^i_k = t^j_\ell$.

The objective of the *blowing-a-kiss problem* is to find a valid schedule S that minimizes $t(S) = \max_{1 \leq i \leq n} t^i_{n-1}$, i.e., the time until all pairs of agents have looked at each other.

We distinguish two models. In the *asynchronous model*, looking at each other can happen at any time. In the *synchronous model* scans need to be synchronized with $\lfloor n/2 \rfloor$ disjoint pairs being scanned at the same time. More specifically, the schedule has *rounds* $r_1, \ldots, r_N$, where $N = n$ if n is odd, and $N = n - 1$ if n is even. In each round an agent can scan one other agent. If n is odd, in every round one agent does not scan, we say this agent has a *bye* or is a *bye agent*. Each round r_i has a timestamp t_i associated with it with $t_i \leq t_{i+1}$. We define the *distance* between two rounds as $d(r_i, r_j) = |t_j - t_i|$.

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.

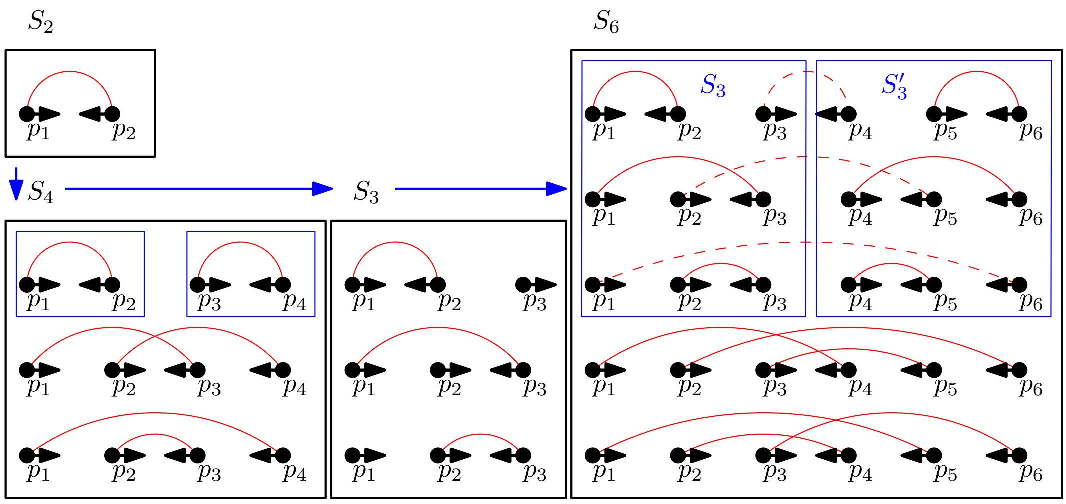

Figure 1 S_6 is constructed via S_2, S_4, S_3. In S_6 the byes from S_3 scan those from S_3' (dashed)

Results. For the case that all agents are on a line, we present a schedule in both models that takes $\pi(\lceil \log n \rceil - 1)$ time, which we prove is optimal. For agents regularly spaced on a circle we present a schedule in the asynchronous model that takes $\pi(\lceil \log n \rceil - 1) + o(1)$ time, which we prove is near optimal. In the synchronous model and for n being a power of 2, we present a schedule that takes at most $2\pi \log n$ time. For the general two-dimensional case we present a schedule in the asynchronous model that takes $\frac{3\pi}{2}\lceil \log n \rceil - \frac{\pi}{2}$ time.

Related work. For the general two-dimensional case, Fekete et al. [4] independently obtained a schedule with the same cost, and additionally prove a lower bound for this case. A related geometric problem is the angular freeze-tag problem [3].

2 Schedules

2.1 Line

In this case the agents $p_1, \ldots, p_n$ are on one line, given from left to right. When facing other agents, an agent has an *orientation*, and it is either oriented to the left or to the right. The time it takes to change the orientation is π.

In the following we construct a schedule S in the synchronous model with $t(S) = \pi(\lceil \log_2 n \rceil - 1)$.The strategy relies on the following lemmata.

▶ **Lemma 2.1.** *Given a schedule S_n for some even n with $t(S_n) = \pi(\lceil \log n \rceil - 1)$, we can construct a schedule S_{n-1} for $n-1$ agents with time $t(S_{n-1}) = \pi(\lceil \log(n-1) \rceil - 1)$ where all bye agents are oriented to the right.*

Proof. Given S_n, remove agent p_n and give a bye to an agent when they would have scanned p_n. This is a valid schedule for $n-1$ agents with time $\pi(\lceil \log n \rceil - 1) = \pi(\lceil \log(n-1) \rceil - 1)$ (since n is even). Since p_n is rightmost, all by agents are oriented to the right. ◀

▶ **Lemma 2.2.** *Given a schedule S_n with $t(S_n) = \pi(\lceil \log n \rceil - 1)$ with all bye agents oriented to the right, we can construct a schedule S_{2n} for $2n$ agents with $t(S_{2n}) = \pi(\lceil \log 2n \rceil - 1)$.*

Proof. Let S'_n be the mirrored schedule of S_n, i.e. the orientation and scans of p_i are swapped with p_{n-i}, and all left orientations change to right and vice versa. First, agents $p_1, \ldots, p_n$ follow the schedule S_n, and agents $p_{n+1}, \ldots, p_{2n}$ follow schedule S'_n simultaneously. If an agent p_i has a bye in S_n, agent p_{2n-i} has a bye in S'_n, and those agents are directed towards each other, and therefore can (and do) scan each other. This removes any byes from the first n rounds, and all unscanned pairs that remain are pairs with one agent in the left half and the other in the right half of $p_1, \ldots, p_{2n}$.

For the remaining rounds, orient the agents in the left half to the right, and in the right half to the left. Consider the graph $G = (P, E)$ on the set P of agents with an edge between any pair of agents that still needs to scan each other. For all $(p, p') \in E$ the agents p and p' are directed towards each other. If n is even, G is a regular bipartite graph of degree n. If n is odd, then each agent had a bye exactly once in S_n, so G is a regular bipartite graph of degree $n - 1$. Since a regular bipartite graph has a 1-factorisation, the final rounds can be scheduled. The first rounds take the same time as S_n, and the final rounds need a single rotation of π, resulting in the claimed time. ◀

Strategy 1. For $n = 2$, the two agents directly scan each other. This is S_2. For $n > 2$, construct S_n recursively from $S_{n/2}$ for even n and from S_{n+1} for odd n. See Fig. 1.

▶ **Theorem 2.3.** *For n agents on a line Strategy 1 constructs a schedule S_n with $t(S_n) = \pi(\lceil \log_2 n \rceil - 1)$ in the synchronous (and asynchronous) model.*

Proof. If $n = 2$, $t(S_n) = 0 = \pi(\lceil \log n \rceil - 1)$ time. If $n > 2$, $t(S_n) = \pi(\lceil \log n \rceil - 1)$ follows inductively by Lemmas 2.1 and 2.2. S_n is also a valid schedule in the asynchronous model, since any schedule in the synchronous model is valid in the asynchronous model. ◀

2.2 Regularly-spaced on circle

In the following the agents $p_1, \ldots, p_n$ are regularly spaced on a circle, given in counter-clockwise order. Consequently, for any p_i, the angular distance between two other consecutive agents is identical (i.e. $\frac{\pi}{n}$). We call this time interval a *step*.

2.2.1 Regularly-spaced, synchronous model with $n = 2^k$.

Strategy 2. Define for every agent p_j the value b^i_j as the value of the i-th bit of j, where j has binary representation $b^k_j \cdots b^1_j$ (not including b^{k+1}_j). The strategy works in phases $1, \ldots, k$, where in phase ℓ agent p_j scans all agents $p_{j'}$ with $b^\ell_j \neq b^\ell_{j'}$ that it has not scanned in an earlier phase in time less than 2π.

Phase ℓ has $2^{k-\ell}$ rounds. In the first round of the phase, every agent p_j is oriented towards and scans agent $p_{j'}$, with

$$j' = \begin{cases} j - 2^{\ell-1} \mod n & \text{if } b^\ell_j = 1 \\ j + 2^{\ell-1} \mod n & \text{if } b^\ell_j = 0 \end{cases} .$$

For the next rounds $i = 1, \ldots, 2^{k-\ell} - 1$, the agents with $b^\ell_j = 1$ rotate clockwise and the agents with $b^\ell_j = 0$ rotate counter-clockwise to scan $p_{j'}$, with

$$j' = \begin{cases} j - 2^{\ell-1}(2i+1) \mod n & \text{if } b^\ell_j = 1 \\ j + 2^{\ell-1}(2i+1) \mod n & \text{if } b^\ell_j = 0 \end{cases} .$$

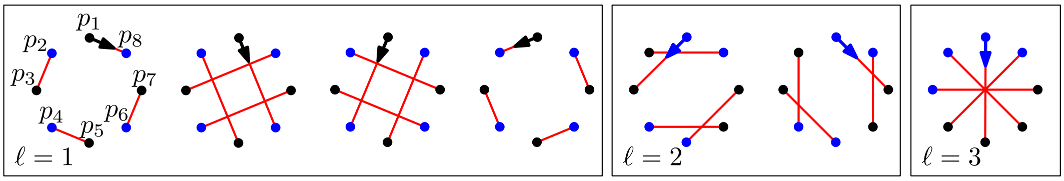

Figure 2 Synchronous schedule for $n = 8$. Agents are blue if the ℓ-th bit is 0, and black otherwise. For $\ell = 1$, p_1 rotates 2 steps clockwise (since $b_1^1 = 1$) in each round. For $\ell = 2$, it rotates 4 steps counter-clockwise (since $b_1^2 = 0$)

Since $j - j' \mod n$ is a multiple of $2^{\ell-1}$ and not a multiple of 2^ℓ, we have $b_j^\ell \neq b_{j'}^\ell$ and so $p_{j'}$ is oriented towards p_j when p_j is oriented towards $p_{j'}$. The angular movement between every two consecutive rounds is equal to $2^\ell \frac{\pi}{n}$, with total movement equal to $\frac{(n-2)\pi}{n}$.

Note that in phase ℓ, every agent p_j scans all other agents $p_{j'}$ where the ℓ-th bit is the smallest bit such that $b_j^\ell \neq b_{j'}^\ell$. So, after k phases, every agent has scanned all other agents.

▶ **Theorem 2.4.** *For $n = 2^k$ agents regularly spaced on a circle Strategy 2 constructs a schedule S with $t(S) \leq 2\pi \log n$ in the synchronous model.*

Proof. For every phase ℓ, the angular movement to complete it is upper bounded by π. The angular movement needed to go from the end state of a step, to the initial state of the next step, is bounded by π as well. Since there are k phases, $t(S) \leq 2\pi k = 2\pi \log n$. ◀

2.2.2 Regularly-spaced, asynchronous model

Strategy 3. The strategy in the asynchronous model works in phases. In each phase, each subset of agents $\{p_i, p_{i+1} \ldots, p_{i+s}\}$ of P that hasn't scanned each other yet is split evenly into a left half $\{p_i, \ldots, p_{i+\lfloor (s-1)/2 \rfloor}\}$ and a right half $\{p_{i+\lfloor (s+1)/2 \rfloor}, \ldots, p_{i+s}\}$. We will scan all pairs between the halves in parallel as follows. See Fig. 3 for an example of a phase.

First, split the left and right halves evenly again into a top and bottom part. W.l.o.g., we assume the subset starts at p_1 and that s is a multiple of 4. Initially, each agent p_j is directed towards p_{s+1-j}. First, scan all pairs between the top left and top right, as follows: for any agent p_j on the top left, rotate towards p_s if p_j and p_s still need to scan each other. Otherwise, rotate towards $p_{3s/4+1}$ until p_j points at $p_{3s/4+1}$. The top right rotates symmetrically. Note that for any $0 \leq j \leq k \leq s/4$, p_{j+1} points to p_{s-k} after $j + k$ steps and p_{s-k} points to p_{j+1} after $k + j$ steps. So, after $s/2 - 1$ steps, all pairs in the top part have been scanned, all agents in the top left point to $p_{3s/4+1}$, and all agents in the top right point to $p_{s/4}$. Symmetrically and in parallel, we can scan all agents in the bottom part after $s/2$ steps, such that all agents in the bottom left point to $p_{3s/4}$, and all agents in the bottom right point to $p_{s/4+1}$.

Next, we rotate each agent in the top left to $p_{3s/4}$ and each agent in the top right to $p_{s/4+1}$, the other agents rotate symmetrically. Then, each agent p_j in the top left waits until it has scanned $p_{3s/4}$, and then rotates towards $p_{s/2-j}$. Note that when p_j and $p_{s/2+1}$ scan each other, p_j is finished for the phase. Additionally, p_j points to $p_{s/2-j}$, the initial orientation for the next phase, after $s/2$ steps. The agents in the bottom half reach the initial orientation for the next phase after $s/2 + n - s$ steps. Now we take each half separately as input for the next phase. After $\lceil \log n \rceil$ phases, all pairs of agents have been scanned.

▶ **Theorem 2.5.** *For n agents regularly spaced on a circle, Strategy 3 constructs a schedule S with $t(S) \leq \pi \frac{n+2}{n}(\lceil \log n \rceil - 1)$ in the asynchronous model.*

Proof. Each phase of Strategy 3, except the last, takes $2(2\lceil\frac{s}{4}\rceil - 1) + n - s \le n + 2$ steps, and the last 0 steps, so Strategy 3 uses $(n+2)(\lceil\log n\rceil - 1)$ steps, taking π/n time each. ◀

2.3 General agents in the plane, asynchronous model

Analogously to kd-tree, the strategy subdivides the set of agents iteratively by vertical and horizontal lines passing through median x- and y-coordinates. In each iteration all agents on one side of the line scan all agents on the other side of the line, resulting in $\lceil\log n\rceil$ iterations.

Strategy 4. We describe one iteration i in which we split the agents along a vertical line, that is when i is odd (refer to Figure 4). The case of even i is analogous. Let P' be the subset that is split, and let A and B be the resulting subsets of P' to the left and right, respectively. Each iteration consists of two stages. In the first stage, all the agents in A rotate by $\pi/2$ to point downwards (direction $3\pi/2$), and all agents in B rotate by $\pi/2$ to point upwards (direction $\pi/2$). In the second, main stage, the points in A and B rotate counter-clockwise to the directions $\pi/2$ and $3\pi/2$ respectively. During this phase all pairs of agents (p, q) with $p \in A$ and $q \in B$ scan each other.

▶ **Theorem 2.6.** *For n agents in the plane, Strategy 4 constructs a schedule S with $t(S) = \frac{3\pi}{2}\lceil\log n\rceil - \frac{\pi}{2}$.*

3 Lower bounds

If a subset of size n' of the agents is nearly collinear, the 1D-analysis gives us that at least $\log n'$ orientation changes taking close to π time must occur, for all agents to see each other. We formalise this idea in the following lemma. It makes use of the fact that at least $\lceil\log n'\rceil$ bipartite graphs are needed to cover the complete graph on n' vertices [5].

▶ **Lemma 3.1.** *Let $r \in \mathbb{R}$, and $Q = \{q_1, \ldots, q_{n'}\}$ be a set of agents in the plane. Suppose Q is ordered such that for all $i \in [n']$ and all $1 \le j < i < k \le n'$, $\measuredangle q_j, q_i, q_k \ge r$. Then, any schedule that scans all pairs of agents in Q takes at least $r \cdot (\lceil\log n'\rceil - 1)$ time.*

Proof. Suppose some schedule scans all pairs of agents in Q within time T. Partition the time interval $[0, T]$ into the intervals $[(i-1)r, ir)$ for $i = 1, \ldots, \lfloor\frac{T}{r}\rfloor$ and the interval $[r\lfloor\frac{T}{r}\rfloor, T]$. Given an interval, consider the graph $G = (Q, E)$ with agents as vertices and an edge between the pairs of agents scanned in the interval. We will show that this graph is bipartite.

Given an interval, call an agent q_i *positive* (resp. negative) in that interval if there is an q_j with $j > i$ (resp. $j < i$) with $(q_i, q_j) \in E$. Since the length of each interval is strictly smaller than r, an agent cannot be both positive and negative in the same interval. Additionally, if $(q_i, q_j) \in E$, exactly one of those agents is positive and the other is negative. So, each edge has an end in the set of positive agents and another end in the set of negative agents. These sets are disjoint, so the graph G is bipartite.

Since all pairs of agents in Q need to be scanned, the union of the bipartite graphs scanned in each interval must be equal to the complete graph. At least $\lceil\log n'\rceil$ bipartite graphs are needed to cover the complete graph on n' vertices, so there must be at least $\lceil\log n'\rceil$ intervals. Since we partitioned $[0, T]$ in $\lfloor\frac{T}{r}\rfloor + 1$ intervals, we have $\frac{T}{r} \ge \lfloor\frac{T}{r}\rfloor \ge \lceil\log n'\rceil - 1$. ◀

▶ **Theorem 3.2.** *Any schedule for the angular blowing-a-kiss problem on a line takes at least $\pi(\lceil\log n\rceil - 1)$ time. The schedule constructed by Strategy 1 is optimal.*

Proof. $Q = P$ and $r = \pi$ satisfy the conditions of Lemma 3.1. ◀

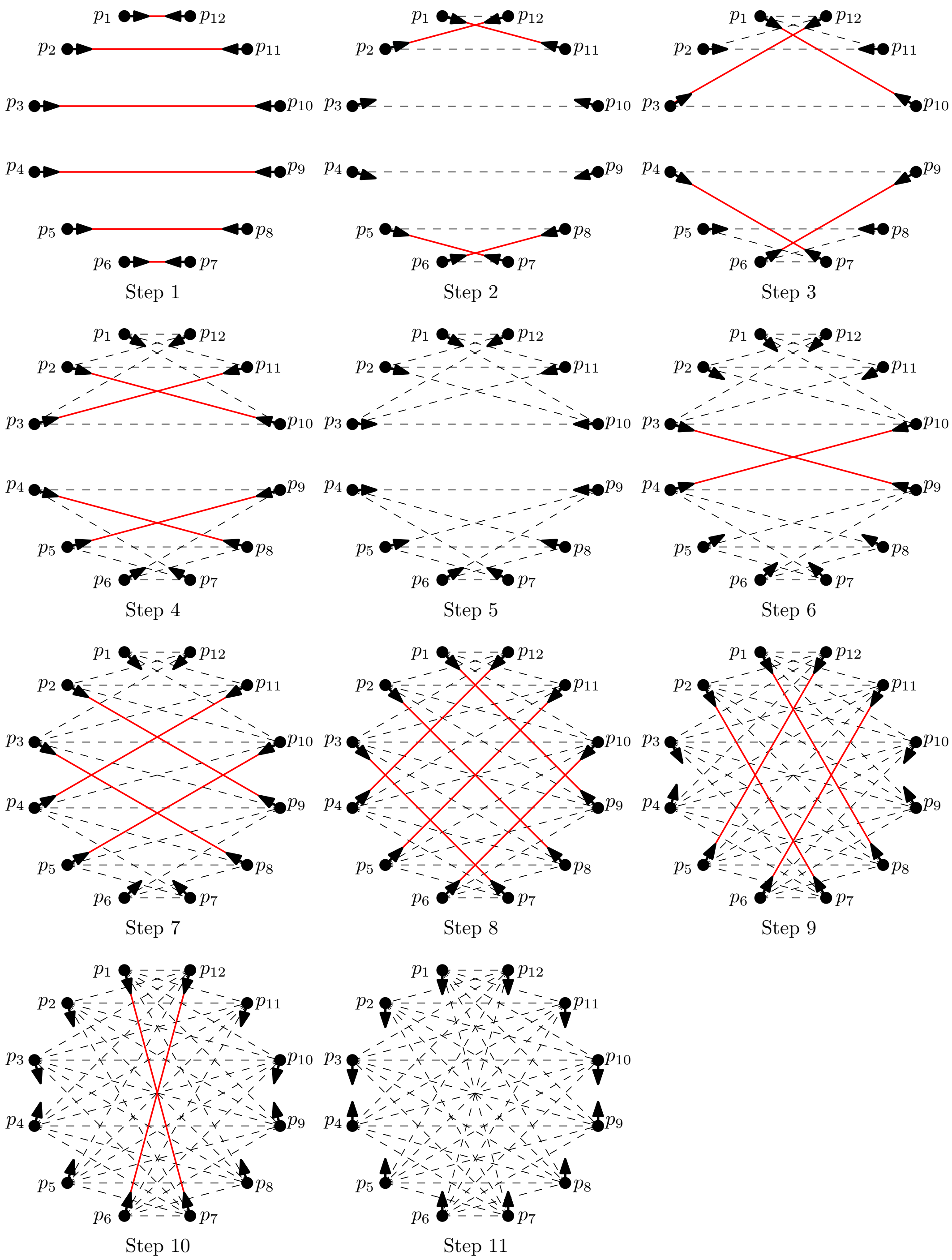

Figure 3 The first phase of Strategy 3 for $n = 12$ has a component of size 12 that is scanned in 10 steps. In step 11, the agents have the correct heading for the next phase. For each step, the edges scanned in that step are colored red, and the edges scanned in an earlier step are dashed.

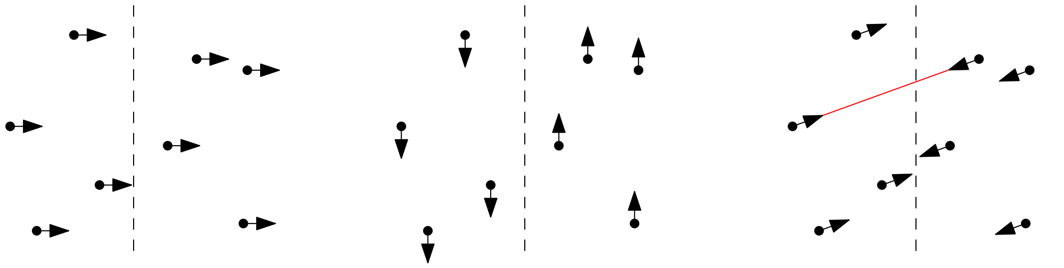

Figure 4 Left: the starting orientation of the agents in iteration i. Middle: the orientation of the agents before the main phase. Right: a snapshot during the main phase when two agents scan each other.

▶ **Theorem 3.3.** *Any schedule for the angular blowing-a-kiss problem with agents regularly-spaced on a circle takes at least* $\pi(1-\frac{1}{\log n})(\log n - \log\log n - 1)$ *time. The schedule constructed by Strategy 3 is asymptotically optimal, i.e. the approximation ratio goes to 1 as* $n \to \infty$*. The approximation ratio of the schedule constructed by Strategy 2 goes to 2 as* $n \to \infty$*.*

Proof. In this configuration, we can take $Q = \{p_1, \ldots, p_k\}$ and $r = (n-k+1)\frac{\pi}{n}$ to satisfy Lemma 3.1, and get a lower bound of $(n-k+1)\frac{\pi}{n}(\lceil \log k \rceil - 1)$ for any $k \in [n]$. Setting $k = \lceil n/\log n \rceil$, we get a lower bound of $\frac{\pi}{n}(n - \lceil n/\log n \rceil + 1)(\lceil \log \lceil n/\log n \rceil \rceil - 1) \geq \frac{\pi}{n}(n - n/\log n)(\log(n/\log n) - 1) = \pi(1 - \frac{1}{\log n})(\log n - \log\log n - 1) \sim \pi \log n$. Strategy 3 takes at most $\pi\frac{n+2}{n}(\lceil \log n \rceil - 1)$ time, so the ratio goes to 1 as $n \to \infty$. The approximation ratio for Strategy 2 is derived analogously. ◀

Acknowledgements. We thank Daniel Cellucci for proposing the problem to us. We also thank Sándor Fekete and Joe Mitchell for fruitful discussions, and Sándor Fekete, Linda Kleist and Dominik Krupke for sharing their paper [4] with us and helpful comments.

References

1 Michael A Bender, Ritwik Bose, Rezaul Chowdhury, and Samuel McCauley. The kissing problem: how to end a gathering when everyone kisses everyone else goodbye. *Theory of Computing Systems*, 54(4):715–730, 2014.

2 Daniel Cellucci. Personal communication, 2018. Distributed Spacecraft Autonomy Project, NASA Ames Research Center, USA.

3 Sándor Fekete and Dominik Krupke. Beam it up, Scotty: Angular freeze-tag with directional antennas. In *Extended Abstracts of 34th European Workshop on Computational Geometry*, 2018.

4 Sándor P. Fekete, Linda Kleist, and Dominik Krupke. Minimum scan cover with angular transition costs. In *Proc. 36th International Symposium on Computational Geometry (SoCG 2020)*, 2020. To appear.

5 Peter C Fishburn and Peter L Hammer. Bipartite dimensions and bipartite degrees of graphs. *Discrete Mathematics*, 160(1-3):127–148, 1996.