

Anonymous Self-Stabilising Localisation via Spatial Population Protocols

Leszek Gąsieniec

University of Liverpool

Łukasz Kuszner

lukasz.kuszner@ug.edu.pl

University of Gdańsk

Ehsan Latif

University of Georgia

Ramviyas Parasuraman

University of Georgia

Paul Spirakis

University of Liverpool

Grzegorz Stachowiak

University of Wrocław

Research Article

Keywords: Population Protocols, Distributed Localisation, Spatial Queries, Self-Stabilisation

Posted Date: April 21st, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-9368524/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License. [Read Full License](#)

Additional Declarations: No competing interests reported.

Anonymous Self-Stabilising Localisation via Spatial Population Protocols

Leszek Gąsieniec¹, Łukasz Kuszner^{2*}, Ehsan Latif³, Ramvijas Parasuraman³,
Paul Spirakis¹, Grzegorz Stachowiak⁴

¹University of Liverpool, Liverpool, United Kingdom.

^{2*}University of Gdańsk, Gdańsk, Poland.

³University of Georgia, Athens, United States of America.

⁴University of Wrocław, Wrocław, Poland.

*Corresponding author(s). E-mail(s): lukasz.kuszner@ug.edu.pl;

Abstract

In the *distributed localisation problem* (DLP), n anonymous robots (agents) $A_0, \dots, A_{n-1}$ are located at arbitrary points $p_0, \dots, p_{n-1} \in S$, where S is a Euclidean space. Initially, each agent A_i operates within its own coordinate system in S , which may be inconsistent with those of other agents. The primary goal in DLP is for agents to reach a consensus on a unified (jointly agreed) coordinate system, in which all agents receive unique *labels* (coordinates) that accurately reflect the relative distances between all points $p_0, \dots, p_{n-1}$ in S . Extensive research on DLP has primarily focus on the feasibility and complexity of achieving consensus when agents have limited access to inter-agent distances, often due to missing or imprecise data. In contrast, this paper proposes a minimalist, computationally efficient distributed computing model where agents can query any pairwise relative positions, if needed. Specifically, we introduce a novel variant of population protocols, referred to as the *spatial population protocols* model. In this variant each agent can memorise one or a fixed number of coordinates, and when agents A_i and A_j interact, they can not only exchange their current knowledge but also either determine the distance d_{ij} between them in S (distance query model) or obtain the vector $\vec{v}_{ij}$ spanning points p_i and p_j (vector query model).

We propose and analyse several distributed localisation protocols, including:

1. *Leader-based localisation protocol with distance queries* We propose and analyse two leader-based localisation protocols that stabilise silently in $\mathcal{O}(n)$ time. These protocols leverage an efficient solution to the novel concept of *multi-contact epidemic*, a natural generalisation of the core communication tool in population protocols, known as the *one-way epidemic*.
2. *Self-stabilising leader localisation protocol with distance queries* We show how to effectively utilise a leader election mechanism within the leader-based localisation protocol to get a DLP protocol that self-stabilises silently in time $\mathcal{O}(n(\log n/n)^{1/(k+1)} \log n)$ in k -dimensions.
3. *Self-stabilising localisation protocol with vector queries* We propose and analyse an optimally fast DLP protocol which self-stabilises silently in $\mathcal{O}(\log n)$ time.

Keywords: Population Protocols, Distributed Localisation, Spatial Queries, Self-Stabilisation

A preliminary version of this work appeared in the ISAAC 2025 proceedings.

1 Introduction

Location services are crucial for modern computing paradigms, such as pervasive computing and sensor networks. While manual configuration and GPS can determine node locations, these methods are impractical in large-scale or obstructed environments. Recent approaches use network localisation, where beacon nodes with known positions enable other nodes to estimate their locations via distance measurements. Key challenges remain, including determining conditions for unique localisability, computational complexity, and deployment considerations. In the *distributed localisation problem* (DLP), n anonymous robots (agents) $A_0, \dots, A_{n-1}$ located at arbitrary points $p_0, \dots, p_{n-1} \in S$, where S is a Euclidean space. Initially, each agent A_i operates within its own coordinate system in S , which may be inconsistent with those of other agents. The primary goal in DLP is for agents to position themselves by adopting labels (coordinates) in a unified (jointly agreed) coordinate system that accurately reflects the relative distances between all points locations $p_0, \dots, p_{n-1}$ in S .

A network’s unique localisability depends on its graph’s combinatorial properties and the number of anchors (agents with known locations). Graph rigidity theory [1–4] states that a planar network is uniquely localisable with at least three anchors and a globally rigid graph. Global rigidity is rare in non-dense networks, though parts may be rigid, leaving some node positions indeterminate [1]. The graph embedding problem, determining if a weighted graph can be embedded in the plane to match edge weights, is strongly NP-hard [5], even for globally rigid graphs [1]. In sensor networks, modelled as unit disk graphs (adjacent nodes within distance r), the unit disk graph reconstruction problem, ensuring adjacent nodes are within r and non-adjacent nodes are farther apart, is also NP-hard [6]. No efficient algorithm exists for localisation unless $P = NP$, even for unique reconstructions [6]. Despite distributed computing models like population protocols [7], localisation remains unstudied in these frameworks.

Distributed localisation is also vital for *robotic systems*, enabling autonomous spatial positioning for navigation, mapping, and multi-robot coordination [8]. It encompasses centralised systems, where a central server computes positions, offering accuracy but limited scalability [9], and distributed systems, where robots independently or collaboratively localise, enhancing resilience but increasing complexity [10, 11]. Leader-based methods use designated robots as reference points [12], while leaderless approaches, leveraging probabilistic [13], geometric [14], or graph-based models [10], excel in decentralised, fault-tolerant settings [15]. Tools like Kalman filters [16], particle filters [9], and graph rigidity theory [17] improve accuracy and efficiency.

A strongly related variant of the localisation problem, known as the *n-point location problem*, has been studied in more centralised setting [18, 19]. This problem was motivated by and is closely related to the computation of relative positions of markers on a DNA string [20–22]. In the *n-point location problem*, the objective is to design one or more rounds of pairwise distance queries between points to determine the relative distances among all points. In [18], a one-round deterministic strategy is presented that utilises $\frac{8n}{5}$ distance queries, along with a proof that any one-round strategy requires at least $\frac{4n}{3}$ distance queries. The same work also introduces a simple two-round deterministic strategy that uses $\frac{3n}{2}$ queries. An alternative two-round randomised approach, which requires only $n + O\left(\frac{n}{\log n}\right)$ queries and solves the *n-point location problem* with high probability, is described in [19].

1.1 Spatial population protocols

This paper presents a minimalist, efficient distributed computing model where randomly paired agents can query their relative positions upon meeting. Our focus is on achieving *anonymity* while maintaining high time efficiency and minimal use of network resources, including limited local storage (agent state space) and communication. To meet these goals, we introduce a new variant of population protocols, referred to as the *spatial population protocols* model.

The *population protocol model* originates from the seminal work by Angluin et al. [23]. This model provides tools for the formal analysis of *pairwise interactions* between indistinguishable entities known as *agents*, which have limited storage, communication, and computational capabilities. In a nutshell, *population protocols* comprise three major components: (1) a *population of agents* $A_0, \dots, A_{n-1}$, which are simple computational entities with limited states that interact in pairs; (2) *interaction rules* forming a *transition function*, which specify agents update their states based on pairwise encounters; and (3) a (possibly random) *scheduler of interactions*, which determines the sequence of agent interactions,

guiding the system toward a desired output or consensus. In *self-stabilising* protocols, the *initial configuration* of agents' states is arbitrary. By contrast, non-self-stabilising protocols start with a predefined configuration encoding the input of the considered problem. A protocol concludes when it *stabilises* in a *final configuration* of states representing the solution of the considered problem. In the *probabilistic variant* of population protocols, which is also used here, the *random scheduler* selects an ordered pair of agents at each step, designated as the *initiator* and the *responder*, uniformly at random from the entire population. The asymmetry in this pair introduces a valuable source of random bits, which is utilised by population protocols. In this probabilistic setting, besides *efficient state utilisation*, *time complexity* is also a primary concern. It is often measured by the number of interactions, $\mathcal{I}$, required for the protocol to stabilise in a final configuration. More recently, the focus has shifted to *parallel stabilisation time* (or simply *time*), defined as $\mathcal{I}/n$, where n is the population size. This measure captures the parallelism of independent, simultaneous interactions, which is leveraged in *efficient population protocols* that stabilise in time $O(\text{poly log } n)$. All protocols introduced in this paper are *stable* (consistently correct), *stabilise silently* (agent states cease to change post-stabilisation), and guarantee stabilisation time with high probability (whp), defined as $1 - n^{-\eta}$, for a constant $\eta > 0$.

Essential tools and context

Among the primary tools in our localisation protocols, we emphasise the novel concept of *multi-contact epidemic*, see Section 2, a natural extension of the fundamental communication mechanism in population protocols, known as the *one-way epidemic*. Another key tool is *leader election*, a cornerstone of distributed computing that tackles the essential challenges of symmetry breaking, synchronisation, and coordination. In population protocols, the presence of a leader facilitates a more efficient computational framework [24]. However, achieving leader election within this model presents significant difficulties. Foundational results [25, 26] demonstrate that it cannot be solved in sublinear time if agents are restricted to a fixed number of states [27]. Further, Alistarh and Gelashvili [28] introduced a protocol stabilising in $O(\log^3 n)$ time with $O(\log^3 n)$ states. Later, Alistarh *et al.* [29] identified trade-offs between state use and stabilisation time, distinguishing *slowly* ($o(\log \log n)$ states) and *rapidly stabilising* ($O(\log n)$ states) protocols. Subsequent work achieved $O(\log^2 n)$ time whp and in expectation with $O(\log^2 n)$ states [30], later reduced to $O(\log n)$ states using synthetic coins [31, 32]. Recent research by Gašieniec and Stachowiak reduced state usage to $O(\log \log n)$ while retaining $O(\log^2 n)$ time whp [33]. The expected time of leader election was further optimised to $O(\log n \log \log n)$ by Gašieniec *et al.* in [34] and to the optimal time $O(\log n)$ by Berenbrink *et al.* in [35]. In contrast, self-stabilising leader election protocols present unique computational challenges. Notably, it was shown by Cai *et al.* in [36] that such protocols require at least n states, in addition to knowledge of the exact value of n . Alternatively, if loose-stabilisation is allowed (where a leader remains in place for a long but finite time before re-election) an upper bound on n may suffice, see work of Sudo *et al.* [37]. In [38, 39], it was established that any silently self-stabilising leader election protocol has an expected time complexity of at least $\Omega(n)$. Furthermore, Burman *et al.* in [38] present silently self-stabilising protocols with expected time $O(n)$ and with high probability (whp) time $O(n \log n)$, both using $n + \Omega(n)$ states. More recently, the space complexity of $O(n \log n)$ -time leader election self-stabilising whp protocol has been improved to $n + O(\log^2 n)$ in [40] and to $n + O(\log n)$ in [41]. In this work, however, we employ a relatively straightforward leader election mechanism where each agent collects $\log n$ random bits, and a leader is determined by a complete set of $\log n$ 1s. This simple protocol ensures the election of a unique leader with constant probability. When repeated $O(\log n)$ times, this procedure guarantees unique leader election whp. For details see Section 3.

We will also require the following lemma.

Lemma 1. *Consider the sum $X = X_1 + X_2 + \dots + X_n$ of n independent 0-1 random variables and any $\delta > 0$. If the expected value $EX \geq c \log n$ for sufficiently large $c > 0$, then $|X - EX| < \delta EX$ holds whp.*

Proof The following equality holds

$$\Pr(|X - EX| < \delta EX) = \Pr(X > (1 + \delta)EX) + \Pr(X < (1 - \delta)EX)$$

By Chernoff inequalities, for c large enough and any parameter η , we get

$$\Pr(X > (1 + \delta)EX) < e^{-EX\delta^2/(2+\delta)} < e^{-c\delta \log n} < n^{-\eta/2}$$

and

$$\Pr(X < (1 - \delta)EX) < e^{-EX\delta^2/2} < e^{-c\delta^2 \log n/2} < n^{-\eta/2}.$$

Thus, $\Pr(|X - EX| < \delta EX) < n^{-\eta}$. $\square$

Spatial embedding with geometric queries

While population protocols offer an elegant and resilient framework for randomised distributed computation, with the exception of work on *graphical models* [42–46], they typically lack topological (in particular, geometric) embedding. To address this limitation, we propose a *spatial variant* of probabilistic population protocols that extends the state space and transition function to accommodate unspecified agent positions $p_0, \dots, p_{n-1}$ distributed in a k -dimensional Euclidean space S . It is assumed that any $k + 1$ agents' positions span the entire space. We also allow two fundamental *geometric queries*.

During an interaction of two agents A_i and A_j , in addition to comparing their current knowledge, the agents can determine:

- (1) the distance d_{ij} separating them in S , in *distance query model*, and
- (2) vector $\vec{v}_{ij}$ spanning points p_i and p_j , in *vector query model*.

As this is the initial work on spatial population protocols, we assume the random scheduler remains unchanged, with pairs of agents selected uniformly at random for interactions. In future work, it would be worth considering models where the probability of agent interactions vary (e.g., in relation to pairwise relative distances) as well as other geometric queries.

1.2 Novelty and structure of presentation

As previously discussed, the localisation problem is fundamental, with many variants studied extensively. This work connects (distributed) localisation with rigidity theory in random geometric graphs, leveraging the increasingly popular population-protocol model. Surprisingly, little work has examined population protocols with spatial properties, especially those enabling geometric queries. The closest prior work concerns efficient majority and leader-election protocols on toroidal grids [45] and self-stabilising leader election in rings [43, 46]. In this study, we assume agents are distributed in a k -dimensional space, where each interaction, viewed as a pairwise geometric query, results in an exchange of states (knowledge held by agents) augmented by information about the relative positions of the interacting agents. We present the following results.

In Section 2, we propose and analyse two leader-based localisation protocols in the model with distance queries. The first protocol (Algorithms 1) stabilises silently in time $O(n(\log n/n)^{1/(k+1)})$ whp (Theorem 2), in k -dimensional space. For $k = 1$, we present a faster protocol with the stabilisation time $O((n \log n)^{1/3})$ whp (Theorem 3). These two protocols leverage an efficient solution to the novel concept of *multi-contact epidemic* (Section 2.1), a natural generalisation of the core communication tool in population protocols, known as the *one-way epidemic*. In Section 3, we show how to effectively utilise leader election within the leader-based localisation protocol (Algorithm 1) in the model with distance queries. We propose and analyse a DLP protocol that self-stabilises silently in time $O(n(\log n/n)^{1/(k+1)} \log n)$ whp (Theorem 4), in k -dimensional space. Finally, in Section 4, we present surprisingly simple and optimally fast DLP protocol which self-stabilises silently in $O(\log n)$ time whp (Theorem 6), in the model with fixed dimension k in the presence of vector queries.

For simplicity, we assume all geometric computations use real arithmetic. Alternatively, one may assume that agents occupy integer grid points, and that each agent rounds any computed position to the nearest grid point.

2 Leader-based localisation in distance query model

In this section, we discuss two localisation protocols with predefined input stabilising in $o(n)$ time, i.e., after this time labels of all agents become stable. These protocols are non-self-stabilising. We assume that one of the agents starts as the *leader* of the population. The agents' point positions $p_0, \dots, p_{n-1}$ are distributed in a k -dimensional Euclidean space S . It is assumed that any $k + 1$ agents' positions span the entire space. For example, in two-dimensional space, this assumption guarantees that no three positions are collinear. Although our algorithms can be adapted to handle an arbitrary distribution of

agents' positions, the time guarantees of such adaptations would be weaker. The state of an agent can accommodate a fixed (related to k) number of agent positions and distances.

We adopt a symmetric model of communication, which means that when agents A_u and A_v interact, they both gain access to each other's states as well as the distance d_{uv} . The early transitions assigned to the leader are distinct from those of the other agents, as the leader also serves as the initiator of the entire process. Initially, the state of each agent A_u stores a label x_u (representing a hypothetical position in S) and its colour $C[A_u]$. We assume that at the beginning, only the leader is coloured **green**, and each non-leader's colour is set to **blue**. Finally, the leader's label (position in S) is set at the origin of the coordinate system, i.e., this label is used as the anchor in the localisation process. In due course, each agent eventually becomes **green**, adopting a label (position) consistent with all other **green** agents upon doing so. Before presenting the localisation protocol, we introduce a novel concept of *multi-contact epidemic*, which is pivotal to our proposed solution.

2.1 Multi-contact epidemic

We introduce and analyse the process of κ -*contact epidemic* (a multi-contact epidemic with a fixed parameter κ), a natural generalisation of epidemic dynamics in population protocols. In this process, the population initially contains at least κ **green** agents, while the remaining agents are **blue**. A **blue** agent turns **green** after interacting with κ distinct **green** agents. We show that the time complexity of this process is $O(n^{1-1/\kappa} \log n)$, for any fixed integer κ .

We start with two key lemmas that outline the progression of nominating **green** agents in the multi-contact epidemic.

Lemma 2. *The time needed to transition from a configuration with m **green** agents to a configuration with $2m$ **green** agents for $m < n^{1/\kappa} \log n$ is bounded by*

$$(c + \kappa) \frac{n^{1-1/\kappa} (\log n)^{1/\kappa}}{\sqrt{m - \kappa + 1}}$$

with high probability (whp), for some constant $c > 0$.

Proof Assume that the number of **green** agents is exactly m . Consider $\kappa - 1$ successive periods of length $n^{1-1/\kappa} (\log n)^{1/\kappa} / \sqrt{m - \kappa + 1}$ and an additional period of length $cn^{1-1/\kappa} (\log n)^{1/\kappa} / \sqrt{m - \kappa + 1}$. We show that after all these periods, we obtain at least m new green agents whp.

We prove by induction that after the first i periods there are at least n_i agents that had at least i contacts (interactions with distinct **green** agents) for $i = 0, 1, 2, \dots, \kappa - 1$, where $n_0 = n - m$ and for $i > 0$,

$$n_i = m(m-1)(m-2) \cdots (m-i+1) \cdot n \left(\frac{\log n}{n} \right)^{i/\kappa} (m - \kappa + 1)^{-i/2}.$$

Now we prove the inductive step. Assume that after initial $i-1$ periods there are at least n_{i-1} agents with at least $i-1$ contacts. Let X_t be a random variable that equals 1 if, at time t , an agent with $i-1$ contacts experiences a new contact (with its i -th **green** agent), and 0 otherwise. If in time t less than n_i agents had i contacts

$$\Pr(X_t = 1) > 2(m-i+1) \frac{n_{i-1} - n_i}{n(n-1)} > 1.5(m-i+1) \frac{n_{i-1}}{n^2}.$$

Let X be the number of agents with i contacts after the i -th period of length $n^{1-1/\kappa} (\log n)^{1/\kappa} / \sqrt{m - \kappa + 1}$. The expected value of $EX \geq \sum_t EX_t$ is at least

$$1.5(m-i+1) \frac{n_{i-1}}{n^2} \cdot n \cdot \frac{n^{1-1/\kappa} (\log n)^{1/\kappa}}{\sqrt{m - \kappa + 1}} > 1.4n_i.$$

By Lemma 1 this number is at least n_i whp. Thus also the number of agents that had at least i contacts is at least n_i whp.

After $\kappa - 1$ periods of length $n^{1-1/\kappa} (\log n)^{1/\kappa} / \sqrt{m - \kappa + 1}$ there are at least $n_{\kappa-1}$ agents that had at least $\kappa - 1$ contacts. We show that after an additional period of length $cn^{1-1/\kappa} (\log n)^{1/\kappa} / \sqrt{m - \kappa + 1}$ we will have at least m new green agents whp. Let Y_t be a random variable that equals 1 if, at time t , an agent with $\kappa - 1$ contacts experiences a new contact (with its κ -th **green** agent), and 0 otherwise. Note that each time $Y_t = 1$, a new **green** agent is produced. As long as less than m agents became green,

$$\Pr(Y_t = 1) > 2(m - \kappa + 1) \frac{n_{\kappa-1} - m}{n(n-1)} > \frac{n_{\kappa-1}}{n^2}.$$

Let $Y = \sum_t Y_t$. After one extra period of length $cn^{1-1/\kappa}(\log n)^{1/\kappa}/\sqrt{m-\kappa+1}$ the expected value $EY = \sum_t EY_t$ is at least $cm(m-1)\cdots(m-\kappa+1) \cdot m^{-\kappa/2} \log n > cm \log n$. By Lemma 1 this number is at least m whp. Thus, also the number of newly generated **green** agents is at least m whp. $\square$

Lemma 3. *Starting with at least $n^{1/\kappa} \log n$ **green** agents guarantees recolouring all n agents to **green** in time $O(n^{1-1/\kappa})$ whp.*

Proof If there are altogether $n^{1/\kappa} \log n$ **green** agents, then for any **blue** agent one can define a random variable X_t equal 1 if during interaction t this agent interacts with a new **green** agent, and 0 otherwise. We have $\Pr(X_t = 1) > 1.5n^{1/\kappa} \log n/n$. Let $X = \sum_t X_t$. In time $ckn^{1-1/\kappa}$ the value $EX = \sum EX_t > 1.5c\kappa \log n$. By Lemma 1 for c big enough $X \geq \kappa$ whp, i.e., each agent becomes **green** whp. $\square$

Theorem 1. *The stabilisation time of κ -contact epidemic is $O(n^{1-1/\kappa} \log^{1/\kappa} n)$ whp.*

Proof The execution time of κ -contact epidemic is the sum of the times to increase the number of **green** agents from $m = \kappa$ to $m = n$, and can be calculated using Lemmas 2 and 3

$$T = O(n^{1-1/\kappa}) + \sum_{m=\kappa, 2\kappa, 4\kappa, 8\kappa, \dots, n^{1/\kappa} \log n} (c + \kappa) \frac{n^{1-1/\kappa}(\log n)^{1/\kappa}}{\sqrt{m-\kappa+1}} = O(n^{1-1/\kappa} \log^{1/\kappa} n),$$

since we assumed that κ is fixed. $\square$

2.2 Localisation via multi-contact epidemic in k -dimensions

After the initialisation phase, once the leader turns **green** and adopts the origin as its label, the localisation protocol proceeds in two parallel components: Process 1 and Process 2 (see Algorithm 1). In Process 1: *nomination* process (lines 8-15), the leader nominates and labels k agents, s.t., their labels (coordinates) span k -dimensional space in a common coordinate system. The nominated agents also become **green**. In Process 2, the *multilateration* process (lines 16-23), each remaining **blue** agent transitions to **green** after interacting with $k+1$ distinct **green** agents. This sequence of interactions allows every agent to determine its position within the common coordinate system established by the leader. Process 2 relies on a direct application of the κ -contact epidemic with $\kappa = k+1$, as discussed in Section 2.1.

Process 1 The positioning (labelling) of the first k **green** agents requires nomination and approval from the leader, who keeps a private list of the currently nominated **green** agents. In particular, for A_v to become the $(i+1)$ -th **green** agent on the leader's list, it must first interact with all i existing **green** agents, and then meet the leader to verify its list of **green** contacts and obtain the nomination. Upon successful verification, the leader updates its list, and the new **green** agent A_v proceeds to compute its projection onto the subspace spanned by its i **green** predecessors and the leader, as well as its Euclidean distance from this subspace. The first i coordinates of A_v 's label are given by this projection, while the $(i+1)$ -th coordinate (in the newly added dimension) equals its distance from the subspace.

Lemma 4. *In Process 1 of Algorithm 1, the i -th **green** agent is positioned in time $O(n^{1-1/i}(\log n)^{1/i})$ whp.*

Proof Consider $i-1$ successive periods of length $n^{1-1/i}(\log n)^{1/i}$ and an additional period of length $cn^{1-1/i}(\log n)^{1/i}$. We show that after all these periods, whp a new **green** agent is positioned.

We now prove by induction that after the first j periods there are at least n_j agents that had at least j interactions with **green** agents for $j = 0, 1, 2, \dots, i-1$, where $n_0 = n-1$ and for $j > 0$, $n_j = n \left(\frac{\log n}{n} \right)^{j/i}$.

We start with the inductive step. Assume that after initial $j-1$ periods there are at least n_{j-1} agents with at least $j-1$ contacts, i.e., interactions with at least $j-1$ **green** agents. Let X_t be a random variable that equals 1 if, at time t , an agent with $j-1$ contacts has a new contact with its j -th **green** agent, and 0 otherwise. If in time t less than n_j agents had j contacts then

$$\Pr(X_t = 1) > 2 \frac{n_{j-1} - n_j}{n(n-1)} > 1.5 \frac{n_{j-1}}{n^2}.$$

Algorithm 1: Leader-based positioning in k dimensions

```
/* Code executed by  $A_u$  exchanging data from  $A_v$  during interaction */
local state :  $x$  – label (position),  $C$  – colour,  $L$  – list of green agents' (position, distance)
input       :  $L(A_v), C(A_v), x(A_v), d_{uv}$ 
1 begin
2   ——— Initialisation ———
3    $L(A_u) \leftarrow$  empty list
4   if  $u$  is the leader then
5      $C(A_u) \leftarrow$  green;  $x(A_u) \leftarrow (0, \dots, 0)$ 
6   end
7    $C(A_u) \leftarrow$  blue while not all agents are positioned do
8     ——— Process 1: nomination ———
9     if  $A_u$  is the leader  $\wedge C(A_v) = \text{blue} \wedge |L(A_u)| = |L(A_v) \setminus (x(A_v), d_{uv})| < k$  then
10      append  $(L(A_u), (x(A_v), d_{uv}))$ 
11    end
12    if  $A_v$  is the leader  $\wedge C(A_u) = \text{blue} \wedge |L(A_u) \setminus (x(A_v), d_{uv})| = |L(A_v)| < k$  then
13       $C(A_u) \leftarrow$  green
14       $x(A_u) \leftarrow$  a position consistent with  $L(A_u)$  and  $(x(A_v), d_{uv})$ 
15    end
16    ——— Process 2: multilateration ———
17    if  $C(A_u)$  is blue  $\wedge C(A_v) = \text{green} \wedge (A_v, d_{uv}) \notin L(A_u)$  then
18      append  $(L(A_u), (x(A_v), d_{uv}))$ 
19      if  $|L(A_u)| = k + 1$  then
20         $C(A_u) \leftarrow$  green
21         $x(A_u) \leftarrow$  the exact position calculated using  $L(A_u)$ 
22      end
23    end
24  end
25 end
```

Now we estimate the number of agents X with j contacts after j -th period of length $n^{1-1/i}(\log n)^{1/i}$. The expected value $EX \geq \sum_t EX_t$ is at least

$$1.5 \frac{n_{j-1}}{n^2} \cdot n \cdot n^{1-1/i} (\log n)^{1/i} > 1.4n_j.$$

By Lemma 1 this number is at least n_j whp. Thus also the number of agents that had interactions with at least j green agents is at least n_j whp.

Furthermore, after $i - 1$ periods there are at least $n_{i-1} = (n/\log n)^{1/i} \log n$ agents that experienced $i - 1$ contacts. Consider an extra period of length $cn^{1-1/i}(\log n)^{1/i}$. Let Y_t be a random variable that equals 1 if, at time t , an agent with $i - 1$ contacts has a contact with the leader, and 0 otherwise. If in time t none of these agents had interaction with the leader yet, we get

$$\Pr(Y_t = 1) > 2 \frac{n_{i-1}}{n^2}.$$

Let Y be the number of agents with $i - 1$ contacts that in extra period of length $n^{1-1/i}(\log n)^{1/i}$ had an interaction with the leader. The expected value of $EY = \sum_t EY_t$ is at least

$$2 \frac{n_{j-1}}{n^2} \cdot n \cdot cn^{1-1/i} (\log n)^{1/i} = 2c \log n.$$

And by Lemma 1 this number is at least 1 whp. □

Process 2 It is sufficient for any remaining agent to interact with $k + 1$ distinct **green** agents to determine its position in the coordinate system established by the leader and thereby become **green**.

We conclude with the following theorem.

Theorem 2. Algorithm 1 solves localisation and stabilises silently in time $O(n(\log n/n)^{1/(k+1)})$ whp.

Proof In Process 1, the protocol allocates *the origin* label to the leader and determines the initial set of $k + 1$ **green** agents in time $O(n(\log n/n)^{1/k})$ whp (Lemma 4). In Process 2, the protocol computes labels (positions) of all remaining agents via $(k + 1)$ -contact epidemic in time $O(n(\log n/n)^{1/(k+1)})$ (Theorem 1). $\square$

2.3 Faster localisation on the line

In this section, we demonstrate how to enhance performance of Algorithm 1 on the line, i.e., within a linear space S . While we focus on one dimension, our observations can be utilised also in higher dimensions. In what follows, we propose an alternative localisation protocol which positions agents not only by using 2-contact epidemic but also utilising interactions between agents with a single **green** contact on their list, coloured as **greenish**. In the new algorithm we take advantage of the fact that there are only two types of interactions between **greenish** agents not leading to positioning of both of them. The first type refers to interactions between agents that became **greenish** via contacting the same **green** agent. In the second, each of the two interacting **greenish** agents is at the same distance and on the same side of their unique **green** contact, rendering positioning impossible. We show that these types of interactions do not contribute significantly to the total time of the stabilisation.

Lemma 5. *The time needed to increase the number of **green** agents from m to $2m$, for $m \in [2, n^{0.9}]$, is $O((n \log n/m)^{1/3})$ whp.*

Proof First we consider an initial period I_1 of length $(n \log n/m)^{1/3}$. The average number of **greenish** agents produced by m **green** agents in time $(n \log n/m)^{1/3}$ is $(m^2 n \log n)^{1/3}$. By Lemma 1 this number is at least $0.9(m^2 n \log n)^{1/3}$ whp. For any given **green** agent A_u , one can define a subset S_u of **greenish** agents originating from contact with agents other than A_u . The cardinality of S_u is on average at least $0.9(m - 1)(n \log n/m)^{1/3}$. By Lemma 1 this number is greater than $0.8(m - 1)(n \log n/m)^{1/3}$ whp. For a given agent A_v , which turned **greenish** after contacting A_u , there are fewer than $2m$ **greenish** agents in S_u with whom no interaction leads to the positioning of A_v , unless the number of **green** agents exceeds $2m$ (they are the second kind of non-positioning agents). Let us consider any set Z of at most $2m$ agents located at points that are translations of the positions of **green** agents by a fixed vector. The expected number of **greenish** agents belonging to Z is at most $n \cdot (n \log n/m)^{1/3} \cdot (2m)^2/n^2 = 4(m^2 n \log n)^{1/3}m/n$. So by Chernoff bound this number is at most $0.1(m^2 n \log n)^{1/3}$ whp.

Thus for a given **greenish** agent A_v , the number of other **greenish** agents with whom interactions position A_v is whp at least

$$0.8(m - 1) \left(\frac{n \log n}{m} \right)^{1/3} - 0.1(m^2 n \log n)^{1/3} > 0.6(m^2 n \log n)^{1/3}.$$

Let I_2 be a time interval of length $c(n \log n/m)^{1/3}$ that follows immediately after I_1 . The probability that an interaction t between two **greenish** agents is the one that positions them and makes them **green** is at least $0.6(m^2 n^{-2} \log n)^{2/3}$. An average number of interactions that position pairs of **greenish** agents in period I_2 is $0.6cm \log n$. By Lemma 1 this yields at least m new **green** agents whp. $\square$

Lemma 6. *The time in which the number of **green** agents increases from $n^{0.9}$ to n is $O(n^{0.1} \log n)$.*

Proof Consider all interaction of agent A_u during period $cn^{0.1} \log n$. The probability of having at most one interaction with a **green** agent for A_u is

$$\begin{aligned} \left(1 - \frac{2n^{0.9}}{n^2}\right)^{cn^{1.1} \log n} + cn^{1.1}(\log n)0.9n^2 \left(1 - \frac{2n^{0.9}}{n^2}\right)^{cn^{1.1} \log n - 1} < \\ < c(\log n)e^{c \log n} < e^{2c \log n} = n^{2c \ln 2}. \end{aligned} \quad (1)$$

So, there is $c > 0$, s.t., all agents have at least two interactions with **green** agents during the considered period $cn^{0.1} \log n$ whp. $\square$

Theorem 3. *The stabilisation time of the improved algorithm is $O((n \log n)^{1/3})$ whp.*

Proof The stabilisation time can be bounded by the sum of the chunks of time needed to increase the number of **green** agents m from 2 to n utilising Lemmas 5 and 6

$$T = O(n^{0.1} \log n) + \sum_{m=2,4,8,\dots,n^{0.9}} O((n \log n/m)^{1/3}) = O((n \log n)^{1/3}).$$

□

3 Self-stabilising localisation in k dimensions

We begin with a brief description of the solution. The proposed self-stabilising protocol in k -dimensions is based on iterative *rounds*, with each round utilising three mechanisms: *leader election* incorporated into *leader-based localisation* (the first stage of a round), and a *buffering mechanism* (the second stage of a round) adopted from [41] (Section 5.2).

Algorithm 2: Self-stabilising positioning in k dimensions

input : Agents start in arbitrary states
output : Agents are positioned in the same system of coordinates

1 ——— **Process 1:** ———
2 **begin**
3 Run *leader election*
4 Run Algorithm 1 (*leader-based localisation*)
5 **end**

6 ——— **Process 2:** ———
7 **begin**
8 **if** *anomaly detected* **then**
9 Run *buffering mechanism*
10 Return to Process 1
11 **end**
12 **end**

During leader election, each agent draws $\log n$ random bits, and if none of these bits is 0, the agent proceeds to the actual localisation protocol as a leader. Note that after all agents complete drawing random bits, which takes $O(\log n)$ time, a unique leader is elected with constant probability, see Lemma 7. Regardless of whether a unique leader is elected, the leader-based localisation protocol continues (possibly indefinitely) unless an *anomaly* is detected. Two types of anomalies may arise. The first, *label inconsistency*, occurs only between two **green** agents. This kind of anomaly occurs when either the initial configuration has conflicting labels of **green** agents or multiple leaders were elected during the current round. When all agents are **green** this type of anomaly is detected in time $O(k \log n)$, see Lemma 8. The second anomaly occurs when any non-**green** agent's counter reaches its deadline $O(n(\log n/n)^{1/(k+1)})$, i.e., the expected stabilisation time of Algorithm 1, indicating that the localisation process is not completed on time. Upon detecting any anomaly, a reset signal is initiated and propagated via a simple epidemic in $O(\log n)$ time. This, together with the (collection and) buffering mechanism adopted from [41] (Section 5.2), ensures sufficient time, namely $O(\log n)$, to reset states of all agents, and prepare them for the next round.

Finally, the time of each round is dominated by the expected time of localisation stabilisation $O(n(\log n/n)^{1/(k+1)})$. Also, the probability of stabilisation in a round is constant, primarily driven by the constant probability associated with successful leader election. Thus, after $O(\log n)$ rounds and a total time of $O(n(\log n/n)^{1/(k+1)} \log n)$, the localisation protocol self-stabilises with high probability (whp), see Theorem 4.

Our result demonstrates that self-stabilisation is not only feasible but also efficient, i.e., comparable to Algorithm 1 presented in Section 2. Moreover, although our self-stabilising localisation protocol relies on leader election, it only requires knowledge of $\log n$, whereas self-stabilising leader election requires precise knowledge of n , see [36]. This is not a contradiction because we only need leader election to succeed with constant probability, which together with efficient anomaly detection, discussed in Section 3.3,

and lightly modified buffering mechanism from [41], discussed in Section 3.4, allows us to restart leader election and in turn the localisation protocol, when necessary.

3.1 Memory utilisation

We begin by analysing the memory utilisation of the protocol. The number of states required for leader election and buffering is $O(\log n)$, as shown in Lemma 7 and Lemma 21 of [41], respectively. However, agents occupying states used for leader election and buffering do not participate in the actual localisation process. As a result, the overall state space of the self-stabilising protocol is dominated by that of Algorithm 1, extended by a deadline counter of size $O(n(\log n/n)^{1/(k+1)})$ used by the **blue** agents.

3.2 Leader election

We begin by presenting a suitable leader election protocol that operates in $O(\log n)$ time and uses $O(\log n)$ space. This protocol is executed at the beginning of the first stage of each round and is followed by the main localisation protocol, Algorithm 1, augmented with anomaly detection and deadline counters.

Leader election is done by independently assigning a leader role to each agent with probability $p = \Theta(1/n)$. In this way, the probability of electing exactly one leader is constant and is maximised when $p = 1/n$ and is then approximately $1/e$. In our protocol, each agent independently tosses a symmetric coin $\log n$ times and becomes the leader if it gets heads on all tosses. We are left to describe the protocol implementing this process.

States in this protocol are denoted as pairs (A, i) , in which $A \in \{N, H, T\}$ and $i \in \{1, \dots, \log n, L, F\}$. The first component, A , represents the type of coin toss state: *Neutral* (N), *Head* (H), or *Tails* (T). The second component, i , either serves as a counter indicating progress toward successful leader election, or denotes a status: L for leader and F for follower. On leaving the buffering mechanism (the second stage of the previous round) all agents receive state $(N, 1)$, and the only meaningful interactions in the leader election protocol are:

- The creation of $H : T$ state pairs ensuring that there are consistently the same number of agents with H and T states in the population: $(N, *) + (N, *) \rightarrow (H, *) + (T, *)$.
- A coin toss in which the initiator gets heads: $(*, i) + (H, *) \rightarrow (*, i + 1) + (H, *)$, if $i < \log n$, and $(*, \log n) + (H, *) \rightarrow (*, L) + (H, *)$, otherwise.
- A coin toss in which the initiator gets tails: $(*, i) + (T, *) \rightarrow (*, F) + (T, *)$, if $i \in \{1, \dots, \log n\}$.

Lemma 7. *A unique leader is elected in $O(\log n)$ time with a constant probability.*

Proof Let us assume that at time 0 all agents have already left the buffer and joined the leader election protocol. For as long as the number of agents in $(N, *)$ states exceeds $n/2$ the probability of forming a $H : T$ pair in a given interaction exceeds $1/4$. Thus, the expected time in which the number of agents in $(N, *)$ states falls below $n/2$ is less than 2, coinciding with $2n$ interactions. And whp this time is less than 3. After this time, the probability that an agent with a counter in an interaction makes a coin flip is $1/2n$. Thus, in a time period of $2c \cdot \log n$ it performs at least $c \cdot \log n$ coin tosses on average. From Chernoff's inequality for a sufficiently large constant c it makes at least $c \cdot \log n$ coin tosses whp. From the union bound in total time $O(\log n)$ all agents will perform $\log n$ coin tosses each, which will establish their leader (L) or follower (F) status. $\square$

We still need to explain how to adopt this leader election protocol in the self-stabilising leader-based localisation protocol. Later in Section 3.4, we show that at some point all agents are located in the buffer, and ready to proceed to the next round. When the first agent leaves the buffer, it assumes the state $(N, 1)$, which triggers an epidemic process that informs all other agents in the buffer to adopt the same state $(N, 1)$. Note that all agents participating in the leader election protocol contribute to this epidemic, ensuring that the entire population engages in leader election within $O(\log n)$ time. Upon completing the leader election, agents with confirmed leader or follower status start leader-based localisation protocol adopting **green** and **blue** states, respectively. In addition, they never give up their coin ($H|T|N$) attributes to guarantee fairness of the remaining coin tosses for agents still executing leader election protocol. Thus, after integration, the overall running time of the leader election protocol remains $O(\log n)$.

3.3 Augmented leader-based localisation

On the conclusion of the leader election process leader-based localisation (Algorithm 1) is executed, however, with a few modifications. Namely, the interaction counter with the deadline $O(n(\log n/n)^{1/(k+1)})$ is added to every new **blue** agent. This is to ensure that the first stage does not exceed the expected stabilisation time of Algorithm 1. One also needs to manage anomaly detection. Recall, that two types of anomalies may arise. The first, *label inconsistency*, occurs only between two **green** agents. In particular, when all agents become **green**, this type of anomaly is detected in time $O(k \log n)$, see the lemma below.

Lemma 8. *When all agents are **green**, the anomaly based on labels inconsistency is detected in time $O(k \log n)$ whp.*

Proof It is known that a random graph with nodes embedded in k -dimensional space and $O(k \log n)$ edges chosen uniformly at random is globally rigid whp [4]. This implies that there exists a unique embedding of the graph resulting from the interactions, up to isometries such as rotations, translations, and reflections. Consequently, if the labels of any **green** agents are inconsistent, such inconsistencies will be detected after $O(k \log n)$ interactions. $\square$

The second anomaly occurs when any **blue** agent's counter reaches the deadline. Upon detecting either of the anomalies, a reset signal is triggered initiating the buffering mechanism discussed in the next section.

3.4 Buffering mechanism

The buffering for reset of the self-stabilising localisation protocol is solved in the same way as the buffering in the ranking protocol with $O(\log n)$ additional states from [41]. In this work, the signal for reset is the detection of an anomaly. It causes the state of the agents that performed the anomaly detection to be set to the first state of the buffer: X_1 . The buffer is a line consisting of $2d \log n$ states: $X_1, X_2, \dots, X_{2d \log n}$ for d large enough. We assign **red** colour to the states $X_1, \dots, X_{d \log n}$ and **white** (**green** in [41]) to the remaining buffer states.

We define the following transitions for these states:

- *Progress* on the line: $X_i + X_j \rightarrow X_{i+1} + X_{i+1}$ for $i \leq j$.
- *Reset propagation* by **red** agents: $X_i + A \rightarrow X_1 + X_1$, if $i \leq d \log n$ and A not on the line.
- *Buffer departure* by **white** agents: $X_{2d \log n} + X_{2d \log n} \rightarrow (N, 1) + (N, 1)$, where $(N, 1)$ is the initial state of the leader election phase. Also $X_i + A \rightarrow (N, 1) + A$, where $i > d \log n$ and A is not on the line.

The analysis of the buffering process is summed up in the paper [41] by the following Lemma, proved there

Lemma 9 (Lemma 21 in paper [41]). *There exists $d' > 0$, s.t., for any $d \geq 0$ there exists $c > 0$, for which after at most $c \log n$ time since state X_1 (reset signal) arrival, all agents are in line states X_i with indices $d \log n < i \leq (d + d') \log n$ whp.*

If we substitute our d for d and d' in this lemma we obtain that at time $O(\log n)$ after the first agents appear in state X_1 , all agents are **white**. On the other hand, when we substitute $2d$ for d in the lemma and d for d' , we obtain that at time $O(\log n)$ the agents start to leave the line of the buffer.

We conclude Section 3 with the following theorem.

Theorem 4. *The localisation protocol presented in this section self-stabilises silently in time $O(n(\log n/n)^{1/(k+1)} \log n)$ whp.*

4 Self-stabilising localisation in vector query model

In this section, we adopt the *vector query model*, assuming first that S is a linear space where agents $A_0, \dots, A_{n-1}$ are arbitrarily positioned at (unknown to the agents) points $p_0, \dots, p_{n-1}$, respectively. We use notation $\vec{v}_{ij}$ to denote the vector connecting p_i and p_j , in this case $\vec{v}_{ij} = p_j - p_i$. Additionally, each agent A_i possesses a *hypothetical coordinate* referred to as its *label* x_i . In this variant of population

protocols, during an interaction of A_i with A_r , the initiator A_i learns about vector $\vec{v}_{ir}$ and label x_r . We show that this model is very powerful as it allows design of an optimal $O(\log n)$ time self-stabilising labelling protocol.

Algorithm 3: Vector query positioning in one dimension

```

/* Code executed by initiator  $A_i$  interacting with  $A_r$  */
local state :  $x_i$ 
input       :  $x_r, \vec{v}_{ir}$ 
1 begin
2   |  $x_i \leftarrow \max\{x_i, x_r - \vec{v}_{ir}\}$ 
3 end

```

We start with a trivial fact concerning label updates.

Fact 1. *For each agent A_i its label x_i never decreases during the execution of Algorithm 3.*

$$\text{Let } M = \max_{0 \leq i \leq n-1} (x_i - p_i).$$

Fact 2. *The value of M does not change during the execution of Algorithm 3.*

Proof Consider an effective update of label x_i , where x_i' is the updated value of this label.

$$x_i' = x_r - \vec{v}_{ir} = x_r - (p_r - p_i). \text{ Thus, } x_i' - p_i = x_r - p_r \leq M. \quad \square$$

After this update agent A_i adopts new label $x_i = x_r - p_r$. Let $S_M = \{A_i : x_i - p_i = M\}$.

Fact 3. *For any two agents $A_i, A_j \in S_M$ we have $x_j - x_i = p_j - p_i$.*

Proof Note that $\vec{v}_{ij} = p_j - p_i = x_j - M - (x_i - M) = x_j - x_i. \quad \square$

Thus, the labels of agents in S_M are stable, meaning they align with the coordinate system that will ultimately be agreed upon by all agents.

Fact 4. *After $A_i \notin S_M$ interaction with $A_r \in S_M$, A_i becomes an element of S_M .*

Proof As $A_i \notin S_M$, we get $x_i - p_i < M$. However, as $r \in S_M$ we also get $x_i - p_i < x_r - p_r$, and in turn $x_i < x_r - p_r + p_i < x_r - \vec{v}_{ir}$. Thus, x_i becomes $x_r - \vec{v}_{ir}$, which is equal to M as $r \in S_M. \quad \square$

Theorem 5. *The localisation Algorithm 3 self-stabilises silently in the optimal time $O(\log n)$ whp.*

Proof The membership of agents in S_M is spread via one-way epidemic in time $O(\log n)$ whp. This protocol is silent as after all agents are included in S_M , further label updates are not possible. Ultimately, this protocol is self-stabilising, requiring no initial assumptions about agents' labels, and achieves time optimality of $O(\log n)$, corresponding to the communication bound in population protocols. $\square$

Algorithm 3 can be extended to higher fixed dimensions by applying multiple instances of the one-dimensional protocol to each dimension's coordinates, as shown in Algorithm 4.

Theorem 6. *The localisation Algorithm 4 self-stabilises silently in a fixed k dimensional space in the optimal time $O(\log n)$ whp.*

Proof By directly applying the union bound. $\square$

Algorithm 4: Vector query positioning in fixed k dimensions

```
/* Code executed by initiator  $A_i$  interacting with  $A_r$  */
local state :  $x_i = [x_i[0], \dots, x_i[k-1]]$ 
input       :  $x_r, \vec{v}_{ir}$ 
1 begin
2   foreach  $j \in [0, k-1]$  do
3      $x_i[j] \leftarrow \max\{x_i[j], x_r[j] - \vec{v}_{ir}[j]\}$ 
4   end
5 end
```

5 Concluding remarks

In this paper, we introduce a novel variant of spatial population protocols and explore its applicability to the distributed localisation problem. Any meaningful advances in this problem could pave the way for developing faster and more robust lightweight communication protocols suitable for real-world applications. It could also provide insights into the limitations of what can be achieved in such systems.

Several challenges remain unresolved in this work. Firstly, we do not account for inaccuracies in distance measurements. As no measuring device is perfect, future studies should model measurement errors, which can significantly affect system stability and performance. Our preliminary studies on the fast positioning protocol from Section 4 reveal a phenomenon called label *drifting* in the presence of errors, resembling phase clock behaviour. This drifting can be controlled to achieve near-complete label stabilisation and may support mobility coordination in large robotic agent populations.

Second, we leave the issue of limited communication range unaddressed. In real-world scenarios, agents often cannot communicate with all other agents in the system, which adds further complexity. As mentioned in the introduction, it is well known that limited communication range and arbitrary network topologies can lead to intractable localisation problems in the worst case. Therefore, a promising direction for future research would be to focus on specific classes of network topologies for which lightweight localisation protocols are more likely to be effective.

A third challenge is the development of efficient localisation algorithms for mobile agents. In this case, assumptions about the relative speeds of communication and movement would likely be necessary to ensure that data from previous positions can still be utilised effectively.

Finally, of independent interest are further studies on the computational power of spatial population protocols, both in comparison to existing variants of population protocols and in relation to various geometric problems, types of queries, distance-based biased communication, and other related topics.

References

- [1] Eren, T., Goldenberg, D.K., Whiteley, W., Yang, Y.R., Morse, A.S., Anderson, B.D.O., Belhumeur, P.N.: Rigidity, computation, and randomization in network localization. In: INFOCOM 2004, pp. 2673–2684 (2004)
- [2] Hendrickson, B.: Conditions for unique graph realizations. SIAM Journal on Computing **21**(1), 65–84 (1992)
- [3] Jackson, B., Jordán, T.: Connected rigidity matroids and unique realizations of graphs. J. Comb. Theory B **94**(1), 1–29 (2005)
- [4] Lew, A., Nevo, E., Peled, Y., Raz, O.E.: Sharp threshold for rigidity of random graphs. Bulletin of the London Mathematical Society **55**(1), 490–501 (2023)
- [5] Saxe, J.B.: Embeddability of weighted graphs in k -space is strongly NP-hard. 17th Allerton Conf. Commun. Control Comput., 1979, 480–489 (1979)
- [6] Aspnes, J., Goldenberg, D.K., Yang, Y.R.: On the computational complexity of sensor network localization. In: ALGOSENSORS 2004, pp. 32–44 (2004)

- [7] Flocchini, P., Prencipe, G., Santoro, N. (eds.): Distributed Computing by Mobile Entities, Current Research in Moving and Computing. LNCS, vol. 11340. Springer, Switzerland (2019)
- [8] Yue, Y., Wang, D.: Collaborative Perception, Localization and Mapping for Autonomous Systems vol. 2. Springer, Singapore (2020)
- [9] Latif, E., Parasuraman, R.: Instantaneous wireless robotic node localization using collaborative direction of arrival. *IEEE Internet of Things Journal* (2023)
- [10] Latif, E., Parasuraman, R.: Dgorl: Distributed graph optimization based relative localization of multi-robot systems. In: International Symposium on Distributed Autonomous Robotic Systems, pp. 243–256 (2022). Springer
- [11] Latif, E., Parasuraman, R.: Multi-robot synergistic localization in dynamic environments. In: ISR Europe 2022; 54th International Symposium on Robotics, pp. 1–8 (2022). VDE
- [12] Fareh, R., Baziyad, M., Khadraoui, S., Brahmi, B., Bettayeb, M.: Logarithmic potential field: A new leader-follower robotic control mechanism to enhance the execution speed and safety attributes. *IEEE Access* (2023)
- [13] Xu, M., Snderhauf, N., Milford, M.: Probabilistic visual place recognition for hierarchical localization. *IEEE Robotics and Automation Letters* **6**(2), 311–318 (2020)
- [14] Latif, E., Parasuraman, R.: GPRL: Gaussian processes-based relative localization for multi-robot systems. In: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2024). IEEE
- [15] Wang, S., Wang, Y., Li, D., Zhao, Q.: Distributed relative localization algorithms for multi-robot networks: A survey. *Sensors* **23**(5), 2399 (2023)
- [16] Moreira, A.P., Costa, P., Lima, J.: New approach for beacons based mobile robot localization using Kalman filters. *Procedia Manufacturing* **51**, 512–519 (2020)
- [17] X. Guo, J., Hu, Chen, J., Deng, F., Lam, T.L.: Semantic histogram based graph matching for real-time multi-robot global localization in large scale environment. *IEEE Robotics and Automation Letters* **6**(4), 8349–8356 (2021)
- [18] Damaschke, P.: Point placement on the line by distance data. *Discret. Appl. Math.* **127**(1), 53–62 (2003)
- [19] Damaschke, P.: Randomized vs. deterministic distance query strategies for point location on the line. *Discret. Appl. Math.* **154**(3), 478–484 (2006)
- [20] J. Håstad, J.L. L. Ivansson: Fitting points on the real line and its application to rh mapping. *J. Algorithms* **49**, 42–62 (2003)
- [21] J. Redstone, W.L.R.: Algorithms for a simple point placement problem. In: CIAC 2000. Lecture Notes in Computer Science, vol. 1767, pp. 32–43 (2000)
- [22] Mumey, B.: Probe location in the presence of errors: a problem from dna mapping. *Discrete Appl. Math.* **104**, 187–201 (2000)
- [23] Angluin, D., Aspnes, J., Diamadi, Z., Fischer, M.J., Peralta, R.: Computation in networks of passively mobile finite-state sensors. In: PODC 2004, pp. 290–299 (2004)
- [24] Angluin, D., Aspnes, J., Eisenstat, D.: Fast computation by population protocols with a leader. *Distributed Comput.* **21**(3), 183–199 (2008)

- [25] Chen, H.-, Cummings, R., Doty, D., Soloveichik, D.: Speed faults in computation by chemical reaction networks. In: Kuhn, F. (ed.) DISC 2014, pp. 16–30 (2014)
- [26] Doty, D.: Timing in chemical reaction networks. In: SODA 2014, pp. 772–784 (2014)
- [27] Doty, D., Soloveichik, D.: Stable leader election in population protocols requires linear time. *Distributed Comput.* **31**(4), 257–271 (2018)
- [28] Alistarh, D., Gelashvili, R.: Polylogarithmic-time leader election in population protocols. In: Halldórsson, M.M., Iwama, K., Kobayashi, N., Speckmann, B. (eds.) ICALP 2015, Part II, pp. 479–491 (2015)
- [29] Alistarh, D., Aspnes, J., Eisenstat, D., Gelashvili, R., Rivest, R.L.: Time-space trade-offs in population protocols. In: SODA 2017, pp. 2560–2579 (2017)
- [30] Bilke, A., Cooper, C., Elsässer, R., Radzik, T.: Brief announcement: Population protocols for leader election and exact majority with $O(\log^2 n)$ states and $O(\log^2 n)$ convergence time. In: PODC 2017, pp. 451–453 (2017)
- [31] Alistarh, D., Aspnes, J., Gelashvili, R.: Space-optimal majority in population protocols. In: SODA 2018, pp. 2221–2239 (2018)
- [32] Berenbrink, P., Kaaser, D., Kling, P., Otterbach, L.: Simple and efficient leader election. In: SODA 2018. OASIs, vol. 61, pp. 9–1911 (2018)
- [33] Gašieniec, L., Stachowiak, G.: Enhanced phase clocks, population protocols, and fast space optimal leader election. *J. ACM* **68**(1), 2–1221 (2021)
- [34] Gašieniec, L., Stachowiak, G., Uznański, P.: Almost logarithmic-time space optimal leader election in population protocols. In: SPAA 2019, pp. 93–102 (2019)
- [35] Berenbrink, P., Giakkoupis, G., Kling, P.: Optimal time and space leader election in population protocols. In: STOC 2020, pp. 119–129 (2020)
- [36] Cai, S., Izumi, T., Wada, K.: How to prove impossibility under global fairness: On space complexity of self-stabilizing leader election on a population protocol model. *Theory of Computer Systems* **50**(3), 433–445 (2012)
- [37] Sudo, Y., Ooshita, F., Kakugawa, H., Masuzawa, T., Datta, A.K., Larmore, L.L.: Loosely-stabilizing leader election with polylogarithmic convergence time. *Theoretical Computer Science* **806**, 617–631 (2020)
- [38] Burman, J., Chen, H.-L., Chen, H.-P., Doty, D., Nowak, T., Severson, E.E., Xu, C.: Time-optimal self-stabilizing leader election in population protocols. In: PODC 2021, pp. 33–44 (2021)
- [39] Doty, D., Soloveichik, D.: Stable leader election in population protocols requires linear time. *Distributed Computing* **31**(4), 257–271 (2018)
- [40] Berenbrink, P., Elsässer, R., Götze, T., Hintze, L., Kaaser, D.: Silent self-stabilizing ranking: Time optimal and space efficient. In: ICDCS 2025, pp. 934–944 (2025)
- [41] Gašieniec, L., Grodzicki, T., Stachowiak, G.: Improving efficiency in near-state and state-optimal self-stabilising leader election population protocols. In: Balliu, A., Kuhn, F. (eds.) PODC 2025, pp. 510–520 (2025)
- [42] Beauquier, J., Blanchard, P., Burman, J.: Self-stabilizing leader election in population protocols over arbitrary communication graphs. In: OPODIS 2013, pp. 38–52 (2013)

- [43] Yokota, D., Sudo, Y., Masuzawa, T.: Time-optimal self-stabilizing leader election on rings in population protocols. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **104-A**(12), 1675–1684 (2021)
- [44] Alistarh, D., Rybicki, J., Voitovich, S.: Near-optimal leader election in population protocols on graphs. In: *PODC'22*, pp. 246–256 (2022)
- [45] Alistarh, D., Gelashvili, R., Rybicki, J.: Fast Graphical Population Protocols. In: *OPODIS 2021*, pp. 1–18 (2022)
- [46] Yokota, D., Sudo, Y., Ooshita, F., Masuzawa, T.: A near time-optimal population protocol for self-stabilizing leader election on rings with a poly-logarithmic number of states. In: *PODC 2023*, pp. 2–12 (2023)