Optimal Robot Localization in Trees*

Kathleen Romanik † Sven Schuierer‡

Abstract

The problem of *localization*, i.e. of a robot finding its position on a map, is an important task for autonomous mobile robots. It has applications in numerous areas of robotics ranging from aerial photography to autonomous vehicle exploration. In this paper we present a new strategy for a robot to find its position on a map where the map is represented as a geometric tree. Our strategy exploits to a high degree the self-similarities that may occur in the environment.

We use the framework of competitive analysis to analyze the performance of our strategy. In particular, we show that the distance traveled by the robot is at most $O(\sqrt{n})$ times longer than the shortest possible route to localize the robot, where n is the number of vertices of the tree. This is a significant improvement over the best known previous bound of $O(n^{2/3})$. Moreover, since there is a lower bound of $\Omega(\sqrt{n})$, our strategy is optimal up to a constant factor.

Using the same approach we can also show that the problem of searching for a target in a geometric tree, where the robot is given a map of the tree and the location of the target but does not know its own position, can be solved by a strategy with a competitive ratio of $O(\sqrt{n})$, which is again optimal up to a constant factor.

*The first author is supported by DIMACS. DIMACS is an NSF Science and Technology Center funded under contract STC-88-09648 and also receives support from the New Jersey Commission on Science and Technology. The second author is supported by the Deutsche Forschungsgemeinschaft under grant No. Ot 64/8-1, "Diskrete Probleme", under grants of the Natural Sciences and Engineering Research Council of Canada, from the Information Technology Research Centre of Ontario, and by an NSERC international fellowship.

†DIMACS, Rutgers University, Piscataway, NJ 08855-1179, email: `romanik@dimacs.rutgers.edu`

‡Institut für Informatik, Universität Freiburg, Am Flughafen 17, Geb. 051, D-79110 Freiburg, Fed. Rep. of Germany, e-mail: `schuiere@informatik.uni-freiburg.de`

1 Introduction

In many tasks of autonomous mobile robots it is assumed that the robot has a map of its environment and knows its location on the map. However, in some situations the robot may not know in advance where its correct position on the map is and has to determine it on-line. This is called the *robot localization problem.* Usually it is assumed that this problem can be solved by using sensor data and allowing the robot to move only a small amount. But if the environment consists of many self-similar parts, this approach may not be sufficient.

Although in robotics this issue has been addressed in numerous contexts, ranging from aerial photography to autonomous vehicles for the exploration of landscapes [10, 12, 14, 15], a more rigorous analysis of the problem in a well-defined theoretical framework has only been considered very recently. Here, the environment of the robot is assumed to be a simple or multiply connected polygon P and the robot is assumed to have access to its local visibility polygon V, i.e. all the points that are visible from its position, via a range sensing device. The robot is also assumed to have a compass so that it knows its orientation. The first task of the robot is to determine its possible placements within P; that is, all $p \in P$ such that the visibility polygon of p equals V. Guibas, Motwani, and Raghavan provide a data-structure that allows efficient enumeration of all such positions [7].

If more than one placement exists, then the robot has to travel to certain points of the polygon that allow it to distinguish between the different placements. Ideally it should not travel much farther than necessary. In order to measure the performance of a localization strategy we employ the framework of competitive analysis [13]. If $L(p, P)$ is the length of a shortest path to localize a robot "waking" in polygon P at position p, a strategy S is called *c-competitive* if the length of the path traveled by a robot using strategy S is at most c times $L(p, P)$, for all possible polygons P and points $p \in P$. The value c is called the *competitive ratio* of S.

There have been two approaches to the localization problem. The first by Dudek, Romanik, and Whitesides [6] considers the full complexity of the problem and utilizes a decomposition of the polygon P into visibility cells such that the same set of vertices of P is visible

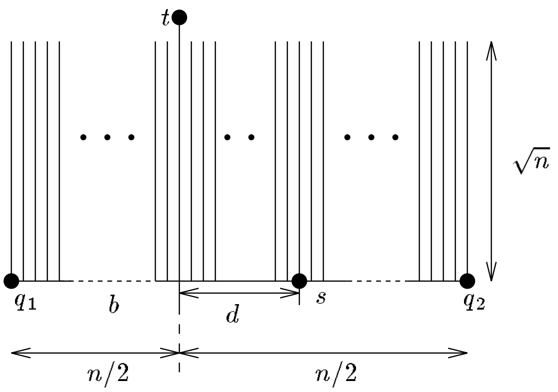

Figure 1: A geometric tree for which every on-line localization strategy is no better than $\Omega(\sqrt{n})$-competitive.

from each point in a cell. This decomposition is used as the underlying structure for a simple strategy where the robot repeatedly travels to the closest point that eliminates at least one of the possible placements. It is easy to show that if there are k possible placements, then the strategy is k-competitive. Furthermore, if $k \leq \sqrt{n}$, where n is the number of vertices of the polygon, then it can be shown that k is the best competitive ratio possible [6].

The second approach, proposed by Kleinberg, leaves aside all concerns raised by the visibility structure of P and abstracts the combinatorial nature of the problem [9]. In this context two types of environments are considered: bounded-degree trees embedded in $\mathbb{E}^d$ (called geometric trees) and rectangle packings in the plane. In these environments the robot has no use of vision other than to know the orientation of all edges incident to its current location. For geometric trees Kleinberg provides a strategy with a competitive ratio of $O(n^{2/3})$, where n is the number of vertices of degree greater than two, and for rectangle packings a strategy with a competitive ratio of $O(n\sqrt{\frac{\log\log n}{\log n}})$, where n is the number of rectangles.

Kleinberg also provides a lower bound of $\Omega(\sqrt{n})$ for the localization problem in geometric trees, which is illustrated by the example in Figure 1. If the true placement s of the robot is at the bottom of the d^{th} spike to the right of spike t, where $\sqrt{n} < d \leq 2\sqrt{n}$, then a localization strategy either has to explore all spikes between t and s or travel to one of the end points q_1 or q_2 of the base b of the tree. In each case the total distance traveled by the robot is $\Omega(n)$ while the shortest path to localize the robot is the path from s to t, which is of length at most $3\sqrt{n}+1$.

In this paper we consider only geometric trees as environments, and we present a localization strategy that has a competitive ratio of $O(\sqrt{n})$. The first and last steps of our strategy are essentially the same as in Kleinberg's $O(n^{2/3})$ strategy. However, in the middle steps we explore periodic patterns in the tree to eliminate all but $O(\sqrt{n})$ possible placements of the robot by traveling no more than $O(\sqrt{n})$ times the length of a shortest path to localize the robot. In view of the example in Figure 1 our strategy is optimal up to a constant factor and, hence, settles the asymptotic complexity of on-line robot localization in trees, thus solving an open problem posed by Kleinberg [9].

A different important problem in robotics is searching for a target in an unknown environment [2, 3, 5, 4, 8, 11]. If a map of the environment and the position of the target on the map are given, but the robot's position on the map is not given, then Figure 1 again provides an example of an $\Omega(\sqrt{n})$ lower bound for the competitive ratio of an on-line strategy. We show that our approach can be used to obtain a strategy that solves the searching problem with a competitive ratio of $O(\sqrt{n})$, which is optimal up to a constant factor.

The paper is organized as follows. In the next section we give formal definitions of the main geometric concepts and list the notation we use. Section 3 contains the main part of the paper where we present our localization strategy. In Section 4 we show how to adapt our strategy to searching for a target.

2 Definitions

In this section we define some of the notation that we use in the paper. The environments that we consider for robot localization are geometric trees.

Definition 2.1 *A* geometric tree $T = (V, E)$ *is a tree embedded into* $\mathbb{E}^d$ *such that each* $v \in V$ *is a point and each* $e \in E$ *is a polygonal path whose end points lie in* V*. The paths of* E *intersect only at points in* V*, and they do not induce any cycles.*

We assume that T has a bounded degree Δ. We say a vertex of T is *non-degenerate* if it has a degree greater than two. The *size* of T, denoted by $|T|$, is then defined as the number of non-degenerate vertices of T. In the remainder of the paper the term vertex is used to refer to a non-degenerate vertex except if explicitly stated otherwise.

We assume that the robot's initial position is located at a vertex of T since the closest vertex v can be reached by performing a two-way spiral search that travels at most nine times the distance from the robot's original location to v [1]. It is the robot's task to determine at which vertex of T it is located by traveling from its starting point in T and collecting enough information to rule out all but one possible start location. We denote the set of all possible start locations of the robot (called *placements*) by P. As in Kleinberg's work, we assume that the robot knows its current orientation, and it has no use of vision other than to know the orientation of all edges incident to its current location.

In order to describe the motion of the robot we assume that the robot has a local coordinate system that is relative to the robot's starting position, i.e. the origin $\mathbf{0}$ of the local coordinate system is at the location where the robot "wakes up". In order to distinguish between points in the global coordinate system and points in the local coordinate system we use the convention that points and paths in the local coordinate system are denoted by over-lined letters, such as $\overline{p}$, $\overline{q}$, $\overline{\mathcal{P}}$, etc. We use standard vector addition and scalar multiplication to denote translations and scalings, i.e. if α and β are reals, v is vector and S is a d-dimensional set, then $\alpha v + \beta S = \{\alpha v + \beta s \mid s \in S\}$.

When dealing with paths in T we use the standard definition of length of a polygonal path (using the L_2 norm) for the length of a path $\mathcal{P}$, which we denote by $\lambda(\mathcal{P})$. We use $sp(T,p,q)$ to denote the unique simple path between two points p and q in T. The *distance* between p and q denoted by $d_T(p,q)$ is defined as the length of $sp(T,p,q)$. Since the robot is often directed to travel along a non-simple path in T (for example, when it explores part of T using depth-first search), for a non-simple path $\mathcal{P}$ with start point s and end point t, we use $\lambda(\mathcal{P})$ to denote the length traveled by the robot and $\lambda_{sp}(\mathcal{P})$ to denote $sp(T,s,t)$. If p and q are two points on $\mathcal{P}$, then we denote the subpath of $\mathcal{P}$ from p to q by $\mathcal{P}_{pq}$. The concatenation of two paths $\mathcal{P}_1$ and $\mathcal{P}_2$ is denoted by $\mathcal{P}_1 * \mathcal{P}_2$, where we assume that the path $\mathcal{P}_2$ is translated so that its start point is the end point of $\mathcal{P}_1$. For $p \in T$, the *ball of radius d around p*, denoted by $B_d(p)$, is defined as the set of all points of T that can be reached by traveling no more than distance d from p.

We say that path $\mathcal{P}_1$ is *isomorphic* to path $\mathcal{P}_2$ if some translation of $\mathcal{P}_1$ is equal to $\mathcal{P}_2$ and all edges adjacent to any vertex on $\mathcal{P}_1$ have the same orientation as the edges adjacent to the corresponding vertex on $\mathcal{P}_2$ (see, for example, Figure 2). We write $\mathcal{P}_1 \equiv \mathcal{P}_2$ in this case.

The following is a summary of some of the notation we use in the paper. Note that the last two items are introduced in the next section.

- T - Geometric tree where robot is located
- n - Number of vertices of T
- $B_d(p)$ - Ball of radius d around p in T
- $sp(T,u,v)$ - Unique shortest path between u and v in T
- $\lambda(\mathcal{P})$; $\lambda_{sp}(\mathcal{P})$ - Length of path $\mathcal{P}$; length of the shortest path from the start to the end point of $\mathcal{P}$
- p_0 - Robot's "wake up" position in T

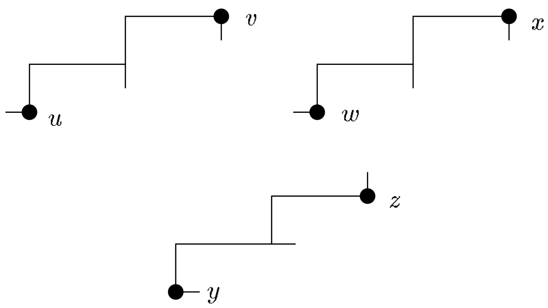

Figure 2: Path $sp(T,u,v)$ is isomorphic to $sp(T,w,x)$ but not to $sp(T,y,z)$.

- P - Set of possible placements of the robot in T
- $\overline{T}$ - Geometric tree T in the robot's local coordinate system
- $\mathbf{0}$ - Origin of $\overline{T}$ (corresponds to p_0 in T)
- $\overline{T}_{ex}$ - Part of $\overline{T}$ explored by the robot
- ϱ - Radius of a ball around p_0 with $B_\varrho(p_0) \subseteq p_0 + \overline{T}_{ex} \subseteq B_{2\varrho}(p_0)$

3 A New Localization Strategy

Our localization strategy consists of four main steps. The following pseudo-code gives a rough impression of how the strategy works.

Strategy *Localize-by-Placement-Separation (LPS)*
Input: A geometric tree $T = (V, E)$ with n vertices;
Output: The position of the robot in T;

Variables: The set of placements P;
The tree $\overline{T}_{ex}$ that has been explored;

$P := V$; $\overline{T}_{ex} := \{\mathbf{0}\}$;
Step 1: Perform a spiral search until either $|\overline{T}_{ex}| > 2\sqrt{n}$ or $|P| = 1$;
Step 2: Identify a critical path of $\overline{T}_{ex}$;
Step 3: Eliminate close placements until $p' \notin p + \overline{T}_{ex}$, for all $p, p' \in P$;
Step 4: Eliminate the remaining placements using a greedy strategy;
end *Localize-by-Placement-Separation*;

It should be noted that in the above strategy Steps 1 and 4 are essentially the same steps as in the strategy proposed by Kleinberg [9]. The only exception is that in Kleinberg's strategy up to $n^{2/3}$ vertices are visited by the Spiral Search in Step 1. After analyzing each step of the strategy we will be able to prove the following theorem.

Theorem 1 *Let T be a geometric tree. Strategy* LPS *achieves a competitive ratio of $O(\sqrt{n})$ for localizing a robot in T.*

3.1 Step 1: Spiral Search

In Step 1 of Strategy LPS the robot performs a spiral search [1] as follows: starting at the origin it performs successive depth-first searches, each time visiting all points within distance d of the origin (where initially d is set to 1) and then doubling d if it has not localized itself. During the search the robot keeps track of the size $|\overline{T}_{ex}|$ of the explored subtree $\overline{T}_{ex}$ and the set P of all possible placements of the robot, i.e., the set of all $p \in T$ with $p + \overline{T}_{ex} \subseteq T$. If $|P| = 1$, then the robot stops its search and outputs $p \in P$ as its position in T.

At the end of the spiral search the robot has seen all points within distance $d/2$ of its wake up position, so for $\varrho = d/2$, $B_\varrho(p_0) \subseteq p_0 + \overline{T}_{ex} \subseteq B_{2\varrho}(p_0)$. If it has not yet eliminated all but one placement, then the length of a shortest path to localize the robot is at least ϱ. It is shown in [9] that the distance traveled by the robot in Step 1 is bounded by $8\Delta|\overline{T}_{ex}|\varrho = O(\sqrt{n}\varrho)$.

3.2 Step 2: Identifying a Critical Path

If after Step 1 there are at least two placements p_1 and p_2 with $p_2 \in p_1 + \overline{T}_{ex}$, then in Step 2 the robot identifies a "critical path" of $\overline{T}_{ex}$ and in Step 3 it travels along an extension of this path eliminating placements until there are no pairs left with $p_2 \in p_1 + \overline{T}_{ex}$. In doing so, the robot travels a distance of at most $O(\sqrt{n}\varrho)$, and at the end at most $\sqrt{n}$ placements remain.

Before describing Step 2 we define some more notions about paths. If $\mathcal{P}$ is a directed simple path and $\alpha \geq 1$, then the *α-repetition of $\mathcal{P}$*, denoted by $\mathcal{P}^\alpha$, is the path formed by concatenating together α copies of $\mathcal{P}$ in succession. If $0 \leq \alpha < 1$, then $\mathcal{P}^\alpha$ is the subpath of $\mathcal{P}$ beginning at its start point with length equal to $\alpha\lambda(\mathcal{P})$. If $\alpha < 0$, then $\mathcal{P}^\alpha$ is the path $\mathcal{P}^{-\alpha}$ translated so that its end point coincides with the start point of $\mathcal{P}$. Note that although $\mathcal{P}$ is a simple path, $\mathcal{P}^\alpha$ may not be (see Figure 3).

If $\mathcal{P}$ is a path such that $\mathcal{P} \equiv \mathcal{Q}^\alpha$, for $\alpha \in [2, \infty)$, then $\mathcal{P}$ is called a *periodic path* and $\mathcal{Q}$ is called a *period of $\mathcal{P}$*. The number α is called the *periodicity of $\mathcal{P}$ w.r.t. $\mathcal{Q}$*. The path $\mathcal{Q}$ is called an *integral period* if α is an integer.

A tree $\mathcal{T}$ is called a *comb-tree* if there is a simple path $\mathcal{Q}$ with start point s and end point t and a simple path $\mathcal{S}$ also with start point s such that $\mathcal{T}$ is given by the union of $\mathcal{Q}^k$ (the *base* of $\mathcal{T}$) and the sets $k_i(t-s)+\mathcal{S}$ (the *spikes* of $\mathcal{T}$), for $1 \leq i \leq m$ where $0 = k_1 < \cdots < k_m = k$ are natural numbers (see Figure 3).

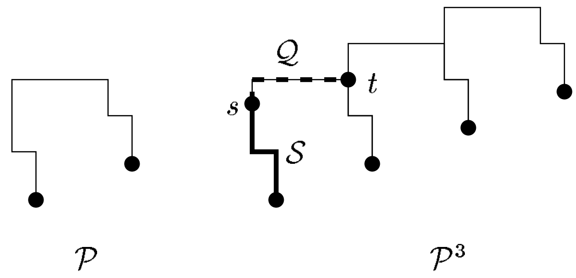

Figure 3: A period that yields a periodic path that is not a simple path. The path $\mathcal{P}^3$ is a comb-tree with base $\mathcal{Q}^3$ and spikes $\mathcal{S}$, $(t-s)+\mathcal{S}$, $2(t-s)+\mathcal{S}$, and $3(t-s)+\mathcal{S}$.

If $U = \{v_1, \ldots, v_k\}$ is a set of vertices of T, then U *induces a periodic path* if there is a path $\mathcal{Q}$ such that $sp(T, v_i, v_{i+1}) \equiv \mathcal{Q}^{k_i}$, for some natural numbers k_i. The set U *induces a comb-tree* if the vertices of U are the leaves of a comb-tree.

Kleinberg [9] proves a crucial lemma stating that all placements within $p + \overline{T}_{ex}$, for a placement p, are on a periodic path or a comb-tree of T. We extend this lemma by proving that after the robot has traveled at most $O(\sqrt{n}\varrho)$ *all* pairs of placements p_1 and p_2 with $p_2 \in p_1 + \overline{T}_{ex}$ are on the *same* periodic path or comb-tree of $\overline{T}_{ex}$. First we state Kleinberg's lemma.

Lemma 3.1 *[9] If p is a placement in P, then the set of all placements that are contained in $(p+\overline{T}_{ex})$ induces either a simple periodic path or a comb-tree.*

If the periodic path that contains all the placements in $(p + \overline{T}_{ex})$ is simple, then we denote this path by $\mathcal{C}_p$ and call it the *critical path of p*. If the placements induce a comb-tree, then the critical path of p is the base of the comb-tree. The critical path $\mathcal{C}_p$ of p corresponds to a path $\overline{\mathcal{C}}_p = -p+\mathcal{C}_p$ in $\overline{T}_{ex}$. One problem that arises is if p_i and p_j are two different placements, then $\overline{\mathcal{C}}_{p_i}$ and $\overline{\mathcal{C}}_{p_j}$ are not necessarily the same, as shown in Figure 4. However, in such cases the robot can efficiently eliminate at least one placement by traveling a distance of $O(\varrho)$, as the next lemma shows.

Lemma 3.2 *If p_1, p'_1, p_2, and p'_2 are placements in P such that $p'_1 \in p_1 + \overline{T}_{ex}$ and $p'_2 \in p_2 + \overline{T}_{ex}$, then either the robot can eliminate p_1 or p_2 by traveling to the point $(p'_1 - p_1) + (p'_2 - p_2)$ in $\overline{T}_{ex}$, or the set $\{\mathbf{0}, p'_1 - p_1, p'_2 - p_2, (p'_1 - p_1) + (p'_2 - p_2)\}$ induces a simple periodic path or a comb-tree in $\overline{T}_{ex}$.*

Proof: Let $\mathcal{P}_i$ be the path from p_i to p'_i, for $i = 1, 2$. Since $p_2 + \overline{T}_{ex}$ is isomorphic to $p'_1 + \overline{T}_{ex}$, there is a path $\mathcal{Q}'_2$ isomorphic to $\mathcal{P}_2$ starting at p'_1 in $p'_1 + \overline{T}_{ex}$ and $\mathcal{P}_1 * \mathcal{Q}'_2 \subset (p_1 + \overline{T}_{ex}) \cup (p'_1 + \overline{T}_{ex})$. If p_0 equals p_1, then the robot can follow the path $\mathcal{P}_1 * \mathcal{Q}'_2$ to its end.

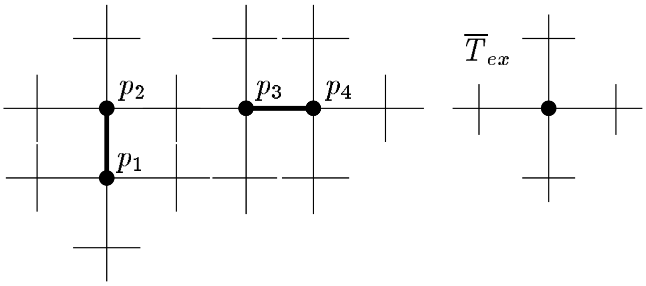

Figure 4: An example where the critical path of p_1 and p_2 is not isomorphic to the critical path of p_3 and p_4.

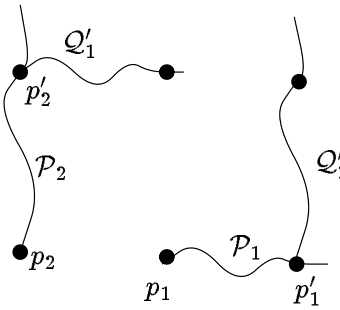

Figure 5: $-p_1 + \mathcal{P}_1 * \mathcal{Q}_2'$ and $-p_2 + \mathcal{P}_2 * \mathcal{Q}_1'$ start and end in the same point.

Similarly, there is a path $\mathcal{Q}_1'$ isomorphic to $\mathcal{P}_1$ starting at p_2' in $p_2'+\overline{T}_{ex}$ and $\mathcal{P}_2 * \mathcal{Q}_1' \subset (p_2+\overline{T}_{ex}) \cup (p_2'+\overline{T}_{ex})$. If p_0 equals p_2, then the robot can follow the path $\mathcal{P}_2 * \mathcal{Q}_1'$ to its end (see Figure 5).

If the robot is able to follow both paths $-p_1+\mathcal{P}_1 * \mathcal{Q}_2'$ and $-p_2+\mathcal{P}_2 * \mathcal{Q}_1'$ from $\mathbf{0}$ to $\overline{q} = (p_1'-p_1)+(p_2'-p_2)$ in $\overline{T}$, then since $\mathcal{P}_1 \equiv \mathcal{Q}_1'$ and $\mathcal{P}_2 \equiv \mathcal{Q}_2'$, by [9, Lemma 3, Claim 1] $\{\mathbf{0}, p_1'-p_1, p_2'-p_2, (p_1'-p_1)+(p_2'-p_2)\}$ induces a simple periodic path or a comb-tree in $\overline{T}_{ex}$. $\square$

If the robot cannot follow path $\mathcal{P}_1 * \mathcal{Q}_2'$ to its end, then p_1 is not a placement, and similarly for path $\mathcal{P}_2 * \mathcal{Q}_1'$ and p_2. Thus, by traveling a distance of at most 16ϱ ($2\varrho + 2\varrho$ to the end of each of the two compound paths and back) the robot can either eliminate one or both of p_1 and p_2, or it can determine that $p_1'-p_1$, $p_2'-p_2$ and $(p_1'-p_1)+(p_2'-p_2)$ are part of the same simple periodic path or comb-tree in $\overline{T}_{ex}$.

In fact, if the robot eliminates one of the placements p_i, $i=1,2$, then it eliminates *all* placements p such that there is another placement at $p+(p_i'-p_i)$; hence, the vertex $\overline{v} = (p_i'-p_i)$ in $\overline{T}_{ex}$ does not correspond to a placement in any translation of $\overline{T}_{ex}$ anymore.

This suggests a method for Step 2 of Strategy LPS. The robot begins with two vertices $\overline{v}_1, \overline{v}_2 \in \overline{T}_{ex}$ that correspond to two placements contained in some translation of $\overline{T}_{ex}$, and it tries to follow the paths $sp(T,\mathbf{0},\overline{v}_1) * sp(T,\mathbf{0},\overline{v}_2)$ and $sp(T,\mathbf{0},\overline{v}_2) * sp(T,\mathbf{0},\overline{v}_1)$ to their end points. If it succeeds, then it finds the periodic path or comb-tree in $\overline{T}_{ex}$ induced by $\{\mathbf{0}, \overline{v}_1, \overline{v}_2, \overline{v}_1+\overline{v}_2\}$; otherwise, at least one of $\overline{v}_1$ and $\overline{v}_2$ is eliminated from consideration. One by one it considers all the other vertices in $\overline{T}_{ex}$, each time eliminating more placements from P or adding them to the periodic path or comb-tree of $\overline{T}_{ex}$ induced by the previously considered vertices. Since the robot travels a distance of at most 16ϱ for each $\overline{v} \in \overline{T}_{ex}$, and $|\overline{T}_{ex}| = 2\sqrt{n}$, it travels no more than $32\varrho\sqrt{n}$ in Step 2.

After the execution of Step 2, let $\overline{U}$ be the set of all vertices $\overline{v} \in \overline{T}_{ex}$ such that there are placements p_1 and p_2 in P with $\overline{v} = p_1 - p_2$. Note that for all $\overline{v}_1$ and $\overline{v}_2$ in $\overline{U}$, the set $\{\mathbf{0}, \overline{v}_1, \overline{v}_2\}$ induces a simple periodic path or a comb-tree by Lemma 3.2. If $|\overline{U}| \geq 3$, then [9, Lemma 3, Claim 2] implies that $\overline{U}$ induces a simple periodic path $\overline{\mathcal{P}}$ or a comb-tree $\overline{\mathcal{T}}$ in $\overline{T}_{ex}$. If $\overline{U}$ induces a comb-tree $\overline{\mathcal{T}}$, then the robot moves to the base of this comb-tree and each $p \in P$ is translated likewise. In this case we denote the base of $\overline{\mathcal{T}}$ by $\overline{\mathcal{P}}$. If we denote the set of translated placements again by $\overline{U}$, then $\overline{U}$ also induces the periodic path $\overline{\mathcal{P}}$ in this case.

Two placements p and p' are called a *critical pair* if $p' \in p + \overline{T}_{ex}$. We denote the set of all critical pairs by $\mathcal{R}^{cr}$. For each critical pair $(p,p') \in \mathcal{R}^{cr}$ the path $-p+sp(T,p,p')$ is an integral period of $\overline{\mathcal{P}}$ by Lemma 3.1. We can show that there is a period of $\overline{\mathcal{P}}$ that divides all paths $-p+sp(T,p,p')$ where (p,p') is a critical pair. The length of this path is the greatest common divisor of the lengths $\lambda(sp(T,p,p'))$.

Lemma 3.3 *If $\mathcal{P}$ is a periodic path and $\{\mathcal{Q}_1,\ldots,\mathcal{Q}_m\}$ is a set of integral periods of $\mathcal{P}$, then there exists a unique period $\mathcal{Q}$ of $\mathcal{P}$, called the* maximal period of $\mathcal{P}$ w.r.t. $\{\mathcal{Q}_1,\ldots,\mathcal{Q}_m\}$ *such that*

(i) $\mathcal{Q}$ is an integral period of $\mathcal{Q}_i$, for all $1 \leq i \leq m$, and

(ii) if $\mathcal{Q}'$ is an integral period of $\mathcal{Q}_i$, for all $1 \leq i \leq m$, then $\mathcal{Q}'$ is also an integral period of $\mathcal{Q}$.

Corollary 3.4 *There is a maximal period $\overline{\mathcal{D}}_{ex}$ of $\overline{\mathcal{P}}$ w.r.t. $\mathcal{R}^{cr}$. It is called the* critical period of $\overline{\mathcal{P}}$.

Note that the above corollary in particular implies that the distance between any critical pair of placements is at least $\lambda(\overline{\mathcal{D}}_{ex})$. With the help of the critical period we now extend the path $\overline{\mathcal{P}}$ that contains all the placements in $\overline{T}_{ex}$. Let α be the largest integer such that $\overline{\mathcal{D}}_{ex}^{\alpha}$ is contained in $\overline{T}_{ex}$. For simplicity of presentation we assume that $\overline{T}_{ex}$ is symmetric w.r.t. $\overline{\mathcal{D}}_{ex}$ and α is also the largest number such that $\overline{\mathcal{D}}_{ex}^{-\alpha}$ is contained in $\overline{T}_{ex}$. (This can easily be achieved by exploring the shorter side.) The path obtained by concatenating $\overline{\mathcal{D}}_{ex}^{-\alpha}$ and

$\overline{\mathcal{D}}_{ex}^{\alpha}$ is called the *critical path of $\overline{T}_{ex}$ w.r.t. P* and is denoted by $\overline{\mathcal{C}}_{ex}$. Note that $\lambda(\overline{\mathcal{C}}_{ex}) \leq 4\varrho$ and that if (p, p') is a critical pair, then the distance between p and p' is at most $\lambda(\overline{\mathcal{C}}_{ex})/2$. Since $\overline{\mathcal{D}}_{ex}$ is also a period of $\overline{\mathcal{C}}_{ex}$, we also call $\overline{\mathcal{D}}_{ex}$ the *critical period of $\overline{\mathcal{C}}_{ex}$ w.r.t. $\mathcal{R}^{cr}$.*

We summarize the situation after Step 2 in the following lemma.

Lemma 3.5 *After Step 2 of Strategy LPS there is a simple periodic path $\overline{\mathcal{C}}_{ex}$ in $\overline{T}_{ex}$ called the* critical path *of $\overline{T}_{ex}$, such that $\overline{p} = p' - p \in \overline{\mathcal{C}}_{ex}$, for all $(p, p') \in \mathcal{R}^{cr}$.*

The critical period has a number of important properties.

Lemma 3.6 *Let Q be a subset of $\mathcal{R}^{cr}$. If $\overline{\mathcal{D}}_{ex}$ is the critical period of $\overline{\mathcal{C}}_{ex}$ w.r.t. $\mathcal{R}^{cr}$ and $\overline{\mathcal{E}}_{ex}$ is the critical period of $\overline{\mathcal{C}}_{ex}$ w.r.t. Q, then $\overline{\mathcal{D}}_{ex}$ is an integral period of $\overline{\mathcal{E}}_{ex}$, i.e. there is an integer $k \geq 1$ such that $\overline{\mathcal{E}}_{ex} = \overline{\mathcal{D}}_{ex}^{k}$.*

3.3 Step 3: Eliminating Close Placements

If after Step 2 there are critical pairs of placements, then in Step 3 the robot explores branches emanating from the critical path $\overline{\mathcal{C}}_{ex}$ until either it has found a periodic tree $\overline{\mathcal{P}}_{ex}$ with $\sqrt{n}$ vertices or there are no more critical pairs. If it succeeds in finding a periodic tree $\overline{\mathcal{P}}_{ex}$ with $\sqrt{n}$ vertices, then there is some point where the tree $\overline{\mathcal{P}}_{ex}^{2\sqrt{n}}$ of $2n$ vertices formed by concatenating together $2\sqrt{n}$ copies of $\overline{\mathcal{P}}_{ex}$ differs from $\overline{T}$. The robot can find one such mismatch by exploring only a small portion of $\overline{\mathcal{P}}_{ex}^{2\sqrt{n}}$. It then propagates this initial mismatch towards $\mathbf{0}$ until all critical pairs are eliminated.

3.3.1 Exploring a Periodic Tree

After the robot has identified the critical path $\overline{\mathcal{C}}_{ex}$, it extends the critical path to a tree that shares the periodicity of $\overline{\mathcal{C}}_{ex}$ w.r.t. $\mathcal{R}^{cr}$. This is done by considering the vertices $\overline{v}_1, \ldots, \overline{v}_l$ of $\overline{T}_{ex}$ in the sequence that the robot encountered them during the last execution of the spiral search. For each vertex $\overline{v}_i$ the robot tries to follow a path that is isomorphic to the path $sp(\overline{T}, \mathbf{0}, \overline{v}_i)$ from each k-repetition $\overline{\mathcal{D}}_{ex}^{k}$ of the critical period. It either succeeds in following this path to its end point, or it finds the first point where this path differs from T.

To describe this strategy more precisely, we first define a few terms.

Definition 3.1 *Let $\overline{q}$ be a point in $\overline{T}$ and $\overline{\mathcal{P}}$ be a path in $\overline{T}$ that starts at point $\overline{s}$ and ends at point $\overline{t}$. If $\overline{q}'$ is the last intersection point of $\overline{\mathcal{P}}$ with $sp(\overline{T}, \overline{s}, \overline{q})$, then $\overline{q}'$ is called the $\overline{\mathcal{P}}$-*ancestor *of $\overline{q}$, denoted by $\overline{\mathcal{P}}$-$anc(\overline{q})$. The path $\overline{\mathcal{P}}_{\overline{s}\overline{q}'} * sp(\overline{T}, \overline{q}', \overline{q}) * sp(\overline{T}, \overline{q}, \overline{q}') * \overline{\mathcal{P}}_{\overline{q}'\overline{t}}$ is the path $\overline{\mathcal{P}}$* augmented by $\overline{q}$ *(see Figure 6).*

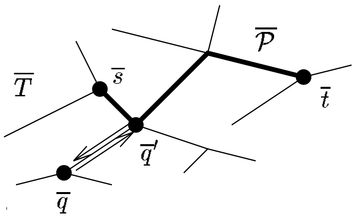

Figure 6: Augmenting the path $\overline{\mathcal{P}}$ by the point $\overline{q}$: The robot follows the path $\overline{\mathcal{P}}$ from $\overline{s}$ to $\overline{q}' = \overline{\mathcal{P}}$-$anc(\overline{q})$, travels to $\overline{q}$ and back, and then follows the rest of $\overline{\mathcal{P}}$.

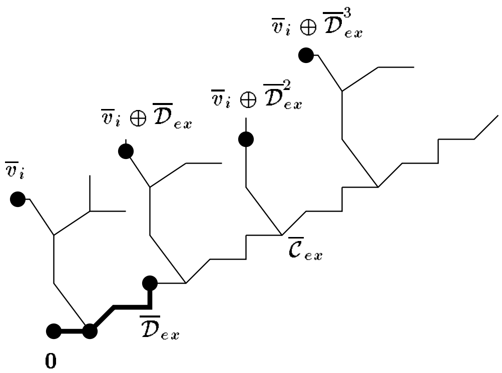

Figure 7: The $\overline{\mathcal{D}}_{ex}^{k}$-neighbours of $\overline{v}_i$, for $1 \leq k \leq 3$. Note that $\overline{v}_i \oplus \overline{\mathcal{D}}_{ex}^{2}$ is not a vertex and $\overline{v}_i \oplus \overline{\mathcal{D}}_{ex}^{3} = \overline{\mathcal{D}}_{ex}^{3} * sp(\overline{T}, \mathbf{0}, \overline{v}_i)$.

The robot explores a non-simple path $\overline{\mathcal{P}}_{ex}$ in $\overline{T}$ that shares the periodicity of $\overline{\mathcal{C}}_{ex}$ w.r.t. $\mathcal{R}^{cr}$. Initially, $\overline{\mathcal{P}}_{ex} = \overline{\mathcal{C}}_{ex}$. The robot then augments $\overline{\mathcal{P}}_{ex}$ by the vertices $\overline{v}_i$. For each vertex $\overline{v}_i$ in $\overline{T}_{ex}$, if $\overline{\mathcal{C}}_{ex}$-$anc(\overline{v}_i)$ is contained in $\overline{\mathcal{D}}_{ex}^{m}$ and $\overline{t}$ is the end point of $\overline{\mathcal{D}}_{ex}^{m-1}$, then the robot attempts to follow the paths $\overline{\mathcal{D}}_{ex}^{k} * sp(\overline{T}, \overline{t}, \overline{v}_i)$, for $0 \leq k < \lambda(\overline{\mathcal{C}}_{ex})/\lambda(\overline{\mathcal{D}}_{ex})$. The last point where the path $\overline{\mathcal{D}}_{ex}^{k} * sp(\overline{T}, \overline{t}, \overline{v}_i)$ is isomorphic to a path in T starting from p_0 is called the *$\overline{\mathcal{D}}_{ex}^{k}$-neighbour* of $\overline{v}_i$ and is denoted by $\overline{v}_i \oplus \overline{\mathcal{D}}_{ex}^{k}$. See Figure 7 for an illustration. The strategy augments $\overline{\mathcal{P}}_{ex}$ by each $\overline{v}_i \oplus \overline{\mathcal{D}}_{ex}^{k}$ and then updates P, $\mathcal{R}^{cr}$, and $\overline{\mathcal{D}}_{ex}$ before examining $\overline{v}_{i+1}$. It halts when either $\mathcal{R}^{cr}$ is empty or $\overline{\mathcal{P}}_{ex}$ contains $\sqrt{n}$ vertices.

Since $|\overline{T}_{ex}| = 2\sqrt{n}$, there is one side of $\overline{\mathcal{C}}_{ex}$ with respect to the origin such that the size of the part of $\overline{T}_{ex}$ that is connected to this side is greater than or equal to $\sqrt{n}$. In order to simplify the presentation we consider in the following only this side of $\overline{\mathcal{C}}_{ex}$ and the part of $\overline{T}_{ex}$ that is connected to it. We denote this half of the explored tree again by $\overline{T}_{ex}$ and also this half of the critical path by $\overline{\mathcal{C}}_{ex}$. Thus we assume that the origin is located at one end point of $\overline{\mathcal{C}}_{ex}$. Note that this implies that if (p, p') is a critical pair, then the distance between p and p' is at most $\lambda(\overline{\mathcal{C}}_{ex})$.

Strategy *Periodic-Tree*
Input: T, P and the explored tree $\overline{T}_{ex}$ with vertices $\overline{v}_1, \ldots, \overline{v}_{\sqrt{n}}$;
Output: A path $\overline{\mathcal{P}}_{ex}$ in $\overline{T}$ that is periodic w.r.t. $\mathcal{R}^{cr}$ and contains $\geq \sqrt{n}$ vertices or $\mathcal{R}^{cr} = \emptyset$;
$P_1 := P$; $\mathcal{R}_1^{cr} := \mathcal{R}^{cr}$; $\overline{\mathcal{P}}_1 := \overline{\mathcal{C}}_{ex}$;
let $\overline{\mathcal{D}}_1$ be the maximal period of $\overline{\mathcal{C}}_{ex}$ w.r.t. $\mathcal{R}_1^{cr}$;
$i := 1$;
while $|\overline{\mathcal{P}}_i| < \sqrt{n}$ **and** $\mathcal{R}_i^{cr} \neq \emptyset$ **do**
let $\overline{v}_i$ be next unvisited vertex among $\overline{v}_1, \ldots, \overline{v}_{\sqrt{n}}$;
let k_i be the periodicity of $\overline{\mathcal{C}}_{ex}$ w.r.t. $\overline{\mathcal{D}}_i$;
let $\overline{\mathcal{P}}_{i+1} := \overline{\mathcal{P}}_i$;
for $j := 0$ **to** $k_i - 1$ **do**
visit $\overline{u}_j := \overline{v}_i \oplus \overline{\mathcal{D}}_i^j$;
augment $\overline{\mathcal{P}}_{i+1}$ by $\overline{u}_j$;
end for;
update the set of placements P_{i+1} using $\overline{\mathcal{P}}_{i+1}$;
update the critical pairs $\mathcal{R}_{i+1}^{cr}$ using P_{i+1};
let $\overline{\mathcal{D}}_{i+1}$ be the maximal period of $\overline{\mathcal{C}}_{ex}$ w.r.t. $\mathcal{R}_{i+1}^{cr}$;
$i := i + 1$;
end while;
$P := P_i$; $\overline{\mathcal{P}}_{ex} := \overline{\mathcal{P}}_i$; $\mathcal{R}^{cr} := \mathcal{R}_i^{cr}$;
end Periodic-Tree;

It is easy to see that Strategy *Periodic-Tree* halts since after at most $\sqrt{n}$ iterations of the while loop, $\overline{\mathcal{P}}_{ex}$ will have $\sqrt{n}$ vertices. Also, it is straightforward to show that after each iteration of the loop, the start and end point of the maximal period $\overline{\mathcal{Q}}_i$ of $\overline{\mathcal{P}}_i$ w.r.t. $\mathcal{R}_i^{cr}$ are the same as the start and end point of the maximal period $\overline{\mathcal{D}}_i$ of $\overline{\mathcal{C}}_{ex}$ w.r.t. $\mathcal{R}_i^{cr}$ (that is, $\lambda_{sp}(\overline{\mathcal{Q}}_i) = \lambda(\overline{\mathcal{D}}_i)$).

To bound the distance that the robot travels during Strategy *Periodic-Tree* we first observe that it travels no more than $O(\varrho)$ to visit each $\overline{\mathcal{D}}_{ex}^k$-neighbour of a vertex $\overline{v}_i$, and it halts when it has visited $\sqrt{n}$ neighbours that are vertices of $\overline{T}$. In order to show that the robot travels no more than $O(\sqrt{n}\varrho)$, it suffices to show that the robot visits no more than $O(\sqrt{n})$ neighbours that are not vertices of $\overline{T}$. Since the periodicity of $\overline{\mathcal{C}}_{ex}$ w.r.t. $\overline{\mathcal{D}}_{ex}$ is never greater than the number of vertices of $\overline{T}_{ex}$, which is $2\sqrt{n}$, it visits no more than $2\sqrt{n}\,\overline{\mathcal{D}}_1^j$-neighbours of $\overline{v}_1$ in the first iteration of the while loop. In each iteration where the robot visits a neighbour that is not a vertex of $\overline{T}$, the length of the critical period $\overline{\mathcal{D}}_i$ at least doubles in length by Lemma 3.6, and thus the number of neighbours visited in the next iteration is at least cut in half. The total number of neighbours visited in all iterations where the robot visits a neighbour that is not a vertex of $\overline{T}$ is thus bounded by $2\sqrt{n} + \sqrt{n} + \frac{1}{2}\sqrt{n} + \frac{1}{4}\sqrt{n} \cdots + 1 < 4\sqrt{n}$.

Lemma 3.7 *The robot travels a distance of $O(\sqrt{n}\varrho)$ during the execution of Strategy* Periodic-Tree*, and after executing it either there are no critical pairs left or it has found a path $\overline{\mathcal{P}}_{ex}$ in $\overline{T}$ that is periodic w.r.t. $\mathcal{R}^{cr}$ and contains $\geq \sqrt{n}$ vertices.*

3.3.2 Extending the Critical Path

After having explored $\overline{\mathcal{P}}_{ex}$ in the local neighbourhood of the wake-up position, there still may exist critical pairs. Since $|\overline{\mathcal{P}}_{ex}| \geq \sqrt{n}$, $|\overline{\mathcal{P}}_{ex}^{2\sqrt{n}}| \geq 2n$ and, thus, there are points, called *mismatches*, where $\overline{\mathcal{P}}_{ex}^{2\sqrt{n}}$ and $\overline{T}$ differ locally. We show that by traveling to at most $\lceil \log n \rceil$ vertices of $\overline{\mathcal{P}}_{ex}^{2\sqrt{n}}$ the robot can find an initial mismatch. This mismatch is then propagated closer to $\mathbf{0}$ until the closest mismatch $\overline{q}_*$ to $\overline{\mathcal{P}}_{ex}$ is identified. Since $\overline{q}_*$ is beyond the end of $\overline{\mathcal{P}}_{ex}$, the maximal period of the part of $\overline{\mathcal{P}}_{ex}^{2\sqrt{n}}$ from $\mathbf{0}$ to $\overline{q}_*$ is longer than $\overline{\mathcal{P}}_{ex}$. Since $d_T(p, p') \leq \lambda(\overline{\mathcal{C}}_{ex}) = \lambda_{sp}(\overline{\mathcal{P}}_{ex})$ for any critical pair (p, p'), this implies that $\mathcal{R}^{cr} = \emptyset$.

Lemma 3.8 *If P is some set of placements in T, then there is a vertex $\overline{v} \in \overline{\mathcal{P}}_{ex}^{2\sqrt{n}}$ such that $p + sp(\overline{\mathcal{P}}_{ex}^{2\sqrt{n}}, \mathbf{0}, \overline{v}) \not\subseteq T$, for at least half of the placements p in P.*

Proof: Let U be the set of pairs $(p, \overline{v})$ with $p \in P$ and $\overline{v} \in \overline{\mathcal{P}}_{ex}^{2\sqrt{n}}$ such that $p + sp(\overline{\mathcal{P}}_{ex}^{2\sqrt{n}}, \mathbf{0}, \overline{v}) \not\subseteq T$. For each $p \in P$, we denote the set of pairs in U with first component p by U_p. Since $|U_p| \geq |\overline{\mathcal{P}}_{ex}^{2\sqrt{n}}| - |T| \geq n$, the size of U is $|U| \geq |P|n$. If $U_{\overline{v}}$ is the set of pairs with second component $\overline{v}$, then $\sum_{\overline{v} \in \overline{\mathcal{P}}_{ex}^{2\sqrt{n}}} |U_{\overline{v}}| = |U| \geq |P|n$. Hence, the average size of the $2n$ sets $U_{\overline{v}}$ is $|P|/2$ and, thus, there is at least one vertex $\overline{v} \in \overline{\mathcal{P}}_{ex}^{2\sqrt{n}}$ with $|U_{\overline{v}}| \geq |P|/2$. $\square$

If we apply Lemma 3.8 repeatedly we obtain the following corollary.

Corollary 3.9 *There are $\mu = \lceil \log |P| \rceil$ vertices $\overline{v}_1, \ldots, \overline{v}_\mu$ in $\overline{\mathcal{P}}_{ex}^{2\sqrt{n}}$ such that, for each $p \in P$, there is a vertex $\overline{v}_{i_p}$ such that $p + sp(\overline{\mathcal{P}}_{ex}^{2\sqrt{n}}, \mathbf{0}, \overline{v}_{i_p}) \not\subseteq T$.*

Let $\overline{v}_1, \ldots, \overline{v}_\mu$ be the vertices of Corollary 3.9, where we assume that the vertices are numbered according to the sequence of their $\overline{\mathcal{C}}_{ex}^{2\sqrt{n}}$-ancestors, i.e. we assume that $\overline{\mathcal{C}}_{ex}^{2\sqrt{n}}$-*anc*$(\overline{v}_i)$ is before $\overline{\mathcal{C}}_{ex}^{2\sqrt{n}}$-*anc*$(\overline{v}_{i+1})$ on $\overline{\mathcal{C}}_{ex}^{2\sqrt{n}}$. Note that these vertices can be computed given $\overline{T}$ and $\overline{\mathcal{P}}_{ex}^{2\sqrt{n}}$ without any additional exploration by the robot. The robot travels along $\overline{\mathcal{C}}_{ex}^{2\sqrt{n}}$ visiting the vertices $\overline{v}_1, \ldots, \overline{v}_\mu$ in sequence until the first vertex $\overline{v}_{i_0}$ is identified for which $p_0 + sp(\overline{\mathcal{P}}_{ex}^{2\sqrt{n}}, \mathbf{0}, \overline{v}_{i_0}) \not\subseteq T$ (see Figure 8 for an illustration). The distance of each vertex $\overline{v}_i$ to $\overline{\mathcal{C}}_{ex}^{2\sqrt{n}}$ is at most 2ϱ; hence, the robot travels a distance

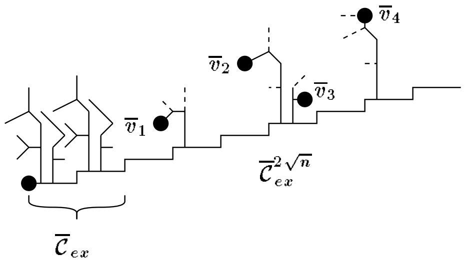

Figure 8: At most $\lceil \log n \rceil$ vertices need to be explored to identify the mismatch in $\overline{v}_4$.

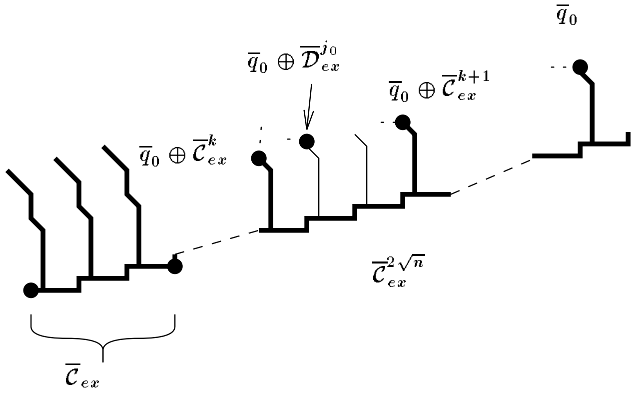

Figure 9: Starting from mismatch $\overline{q}_0$ the closest mismatch found by propagation is $\overline{q}_0 \oplus \overline{\mathcal{D}}_{ex}^{j_0}$. All $\overline{\mathcal{D}}_{ex}$-neighbours of $\overline{q}_0$ between $\overline{q}_0 \oplus \overline{\mathcal{C}}_{ex}^{k}$ and $\overline{q}_0 \oplus \overline{\mathcal{C}}_{ex}^{k+1}$ are explored.

of at most $\lambda(\overline{\mathcal{C}}_{ex}^{2\sqrt{n}}) + 4\lceil \log n \rceil \varrho = (4\sqrt{n} + 4\lceil \log n \rceil)\varrho$ in order to identify $\overline{v}_{i_0}$. We denote by $\overline{q}_0$ the last common point of $sp(\overline{\mathcal{P}}_{ex}^{2\sqrt{n}}, \mathbf{0}, \overline{v}_{i_0})$ and $\overline{T}$.

In order to find the closest mismatch to $\overline{\mathcal{C}}_{ex}$, the robot executes a strategy called *Mismatch-Propagation*. It returns to $\mathbf{0}$ on $\overline{\mathcal{C}}_{ex}^{2\sqrt{n}}$ and on its way visits points that are isomorphic to $\overline{q}_0$. It first travels by multiples of $\overline{\mathcal{C}}_{ex}$ examining neighbours of $\overline{q}_0$ until it finds one that is not a mismatch. It then examines all the neighbours of $\overline{q}_0$ on the last repetition of $\overline{\mathcal{C}}_{ex}$ by visiting neighbours that are multiples of $\overline{\mathcal{D}}_{ex}$ along the path. It finds the closest mismatch to $\mathbf{0}$ among the $\overline{\mathcal{D}}_{ex}^{k}$-neighbours of $\overline{q}_0$ it has examined and repeats the process starting from this point. If the robot reaches $\overline{\mathcal{P}}_{ex}$ before $\mathcal{R}^{cr} = \emptyset$, then it explores the entire part of the tree $\overline{\mathcal{P}}_{ex}^{2\sqrt{n}}$ between this mismatch and the end of $\overline{\mathcal{P}}_{ex}$.

More precisely, if m is the smallest integer such that $\overline{\mathcal{P}}_{ex}^{m}$ contains $\overline{q}_0$, and $\overline{t}$ is the end point of $\overline{\mathcal{C}}_{ex}^{m-1}$, then the $\overline{\mathcal{C}}_{ex}^{k}$-neighbour of $\overline{q}_0$ ($\overline{q}_0 \oplus \overline{\mathcal{C}}_{ex}^{k}$) is the last point where $\overline{\mathcal{C}}_{ex}^{k} * sp(\overline{\mathcal{P}}_{ex}^{2\sqrt{n}}, \overline{t}, \overline{q}_0)$ is isomorphic to a path in $\overline{T}$. Similarly, if l is the smallest integer such that $\overline{\mathcal{Q}}_{ex}^{l}$ contains $\overline{q}_0$, and $\overline{s}$ is the end point of $\overline{\mathcal{D}}_{ex}^{l-1}$, then the $\overline{\mathcal{D}}_{ex}^{k}$-neighbour of $\overline{q}_0$ ($\overline{q}_0 \oplus \overline{\mathcal{D}}_{ex}^{k}$) is the last point where $\overline{\mathcal{D}}_{ex}^{k} * sp(\overline{\mathcal{P}}_{ex}^{2\sqrt{n}}, \overline{s}, \overline{q}_0)$ is isomorphic to a path in $\overline{T}$. The robot first visits the neighbours $\overline{q}_0 \oplus \overline{\mathcal{C}}_{ex}^{k}$ of $\overline{q}_0$ for decreasing k starting from $m-2$ until it finds a neighbour that is not a mismatch.

The robot then explores the neighbours $\overline{q}_0 \oplus \overline{\mathcal{D}}_{ex}^{j}$ of $\overline{q}_0$ between $\overline{q}_0 \oplus \overline{\mathcal{C}}_{ex}^{k}$ and $\overline{q}_0 \oplus \overline{\mathcal{C}}_{ex}^{k+1}$ and finds the smallest j such that $\overline{q}_1 = \overline{q}_0 \oplus \overline{\mathcal{D}}_{ex}^{j}$ is a mismatch (see Figure 9 for an illustration). It then updates P, $\mathcal{R}^{cr}$, and $\overline{\mathcal{D}}_{ex}$ according to what it has explored, and the propagation is repeated starting from $\overline{q}_1$. It continues until either $\mathcal{R}^{cr} = \emptyset$ or $\overline{\mathcal{P}}_{ex}$ is reached. In the latter case, the remainder of the tree $\overline{\mathcal{P}}_{ex}^{2\sqrt{n}}$ from the mismatch to $\overline{\mathcal{P}}_{ex}$ is explored.

Pseudo-code for Strategy *Mismatch-Propagation* is below. We assume for simplicity of presentation that $\overline{\mathcal{D}}_i$ is an integral period of $\overline{\mathcal{C}}_{ex}$ in all iterations of the outer while loop.

Strategy *Mismatch-Propagation*
Input: The tree T, the set of placements P, the periodic paths $\overline{\mathcal{P}}_{ex}$, $\overline{\mathcal{C}}_{ex}$, and the point $\overline{q}_0$;
Output: A set of placements P where $\mathcal{R}^{cr} = \emptyset$;
$P_0 := P$; $\overline{\mathcal{D}}_0 = \overline{\mathcal{D}}_{ex}$;
$\overline{\mathcal{P}}_0 = \overline{\mathcal{P}}_{ex} \cup sp(\overline{T}, \mathbf{0}, \overline{q}_0)$;
$i := 0$;
while $\mathcal{R}^{cr} \neq \emptyset$ **do**
  let $\overline{\mathcal{P}}_{i+1}$ be $\overline{\mathcal{P}}_i$;
  let m be the integer such that $\overline{\mathcal{P}}_{ex}^{m}$ contains $\overline{q}_i$;
  $k := m-2$; visit $\overline{q}_i \oplus \overline{\mathcal{C}}_{ex}^{k}$;
  augment $\overline{\mathcal{P}}_{i+1}$ by $\overline{q}_i \oplus \overline{\mathcal{C}}_{ex}^{k}$;
  while $\overline{q}_i \oplus \overline{\mathcal{C}}_{ex}^{k}$ is a mismatch **do**
    $k := k-1$; visit $\overline{q}_i \oplus \overline{\mathcal{C}}_{ex}^{k}$;
    augment $\overline{\mathcal{P}}_{i+1}$ by $\overline{q}_i \oplus \overline{\mathcal{C}}_{ex}^{k}$;
  end while;
  /* $\overline{q}_i \oplus \overline{\mathcal{C}}_{ex}^{k}$ *is a match,* $\overline{q}_i \oplus \overline{\mathcal{C}}_{ex}^{k+1}$ *is a mismatch* */
  let l be the integer such that $\overline{\mathcal{D}}_i^{l}$ contains the
    $\overline{\mathcal{C}}_{ex}^{2\sqrt{n}}$-ancestor of $\overline{q}_i \oplus \overline{\mathcal{C}}_{ex}^{k}$;
  let π be the periodicity of $\overline{\mathcal{C}}_{ex}$ w.r.t. $\overline{\mathcal{D}}_i$;
  for $j := l$ **to** $l + \pi - 1$ **do**
    visit $\overline{q}_i \oplus \overline{\mathcal{D}}_i^{j}$; augment $\overline{\mathcal{P}}_{i+1}$ by $\overline{q}_i \oplus \overline{\mathcal{D}}_i^{j}$;
  end for;
  let j_0 be $\min\{j \mid \overline{q}_i \oplus \overline{\mathcal{D}}_i^{j}$ is a mismatch$\}$;
  /* $\overline{q}_i \oplus \overline{\mathcal{D}}_i^{l+\pi-1} = \overline{q}_i \oplus \overline{\mathcal{C}}_{ex}^{k+1}$ *is a mismatch* */
  $\overline{q}_{i+1} := \overline{q}_i \oplus \overline{\mathcal{D}}_i^{j_0}$;

update the set of placements P_{i+1} using $\overline{\mathcal{P}}_{i+1}$;
update the critical pairs $\mathcal{R}^{cr}$ using P_{i+1};
let $\overline{\mathcal{D}}_{i+1}$ be the maximal period of $\overline{\mathcal{C}}_{ex}$ w.r.t. $\mathcal{R}^{cr}$;
$i := i + 1$;
end while;
$P := P_i$;
end Mismatch-Propagation;

To show that Strategy *Mismatch-Propagation* is correct we must show that it halts and at the end $\mathcal{R}^{cr} = \emptyset$. Observe that the period of the part of $\overline{\mathcal{P}}_{i+1}$ between $\overline{q}_i \oplus \overline{\mathcal{C}}^k_{ex}$ and $\overline{q}_i \oplus \overline{\mathcal{C}}^{k+1}_{ex}$ inclusive, which the robot explores in one iteration of the outer while loop, determines the new critical period $\overline{\mathcal{D}}_{i+1}$. Since this part of $\overline{\mathcal{P}}_{i+1}$ contains at least one mismatch $(\overline{q}_i \oplus \overline{\mathcal{C}}^{k+1}_{ex})$ and at least one match $(\overline{q}_i \oplus \overline{\mathcal{C}}^k_{ex})$, the new period of $\overline{\mathcal{C}}_{ex}$ w.r.t. $\mathcal{R}^{cr}$is different from $\overline{\mathcal{D}}_i$. By Lemma 3.6 the critical period at least doubles in length. In the extreme case that $\overline{q}_i \oplus \overline{\mathcal{C}}^{k+1}_{ex}$ is the closest mismatch of $\overline{q}_i \oplus \overline{\mathcal{C}}^k_{ex}$ (i.e., if $j_0 = l + \pi - 1$) the new period $\overline{\mathcal{D}}_{i+1}$ equals $\overline{\mathcal{C}}_{ex}$ and the distance between any two placements is longer than $\lambda(\overline{\mathcal{C}}_{ex})$. Hence, $\mathcal{R}^{cr}$ becomes empty and the strategy halts.

Otherwise, after the iteration of the outer while loop the mismatch $\overline{q}_i$ is propagated forward towards $\overline{\mathcal{P}}_{ex}$, and eventually $\overline{\mathcal{P}}_{ex}$ is reached in some iteration i_0. In iteration i_0 all $\overline{\mathcal{D}}_{i_0}$-neighbours between the end of $\overline{\mathcal{P}}_{ex}$ and the last mismatch found are explored. Since the $\overline{\mathcal{D}}_{i_0}$-neighbours of $\overline{q}_{i_0}$ that are contained in $\overline{\mathcal{P}}_{ex}$ have already been visited during the construction of $\overline{\mathcal{P}}_{ex}$, $\overline{\mathcal{P}}_{ex}$ contains only matches and the closest mismatch $\overline{q}_*$ to the origin that is found is located after the end of $\overline{\mathcal{P}}_{ex}$. This again implies that the distance between two placements is longer than $\lambda(\overline{\mathcal{C}}_{ex})$, so $\mathcal{R}^{cr} = \emptyset$ and the strategy halts.

In order to determine the distance traveled by the robot during Strategy *Mismatch-Propagation* we look at the different parts of the strategy. Since $d_0 = d_{\overline{T}}(\mathbf{0}, \overline{\mathcal{C}}^{2\sqrt{n}}_{ex}\text{-}anc(\overline{q}_0)) \leq 2\sqrt{n}\lambda(\overline{\mathcal{C}}_{ex})$, the robot visits no more than $O(\sqrt{n})$ $\overline{\mathcal{C}}^k_{ex}$-neighbours during the execution of the strategy, and it travels no more than $2\sqrt{n}\lambda(\overline{\mathcal{C}}_{ex})$ along $\overline{\mathcal{C}}^{2\sqrt{n}}_{ex}$ visiting these neighbours. Since each neighbour is no farther than 2ϱ from $\overline{\mathcal{C}}^{2\sqrt{n}}_{ex}$ and $\lambda(\overline{\mathcal{C}}_{ex}) \leq 2\varrho$, the robot travels no more than $O(\sqrt{n}\varrho)$ during this part of the strategy.

Since in each iteration of the outer while loop the critical period $\overline{\mathcal{D}}_i$ at least doubles in length by the above argument, the number of $\overline{\mathcal{D}}^j_i$-neighbours between $\overline{q}_{i+1} \oplus \overline{\mathcal{C}}^k_{ex}$ and $\overline{q}_{i+1} \oplus \overline{\mathcal{C}}^{k+1}_{ex}$ visited in the next iteration is at least cut in half. Therefore, using a similar argument to the one used for Strategy *Periodic-Tree*, it can be shown that the robot visits a total of no more than $2\sqrt{n}$ $\overline{\mathcal{D}}^j_{ex}$-neighbours and thus travels no more than $O(\sqrt{n}\varrho)$ during this part of the strategy. Therefore, we have the following lemma.

Lemma 3.10 *The robot travels a distance of $O(\sqrt{n}\varrho)$ during the execution of Strategy* Mismatch-Propagation *and after executing it, there are no critical pairs left.*

3.4 Step 4: Greedy Elimination

Since $2\sqrt{n}$ vertices of T have been visited by the robot at the end of Step 1 and $\mathcal{R}^{cr} = \emptyset$ at the end of Step 3, we can show that at the beginning of Step 4 the number of remaining placements is less than $\sqrt{n}$.

Lemma 3.11 *If $|\overline{T}_{ex}| \geq 2\sqrt{n}$ and $\mathcal{R}^{cr} = \emptyset$, then $|P| < \sqrt{n}$.*

Proof: The proof is by contradiction. Assume that $|P| \geq \sqrt{n}$. Since there are no critical pairs, $P \cap p + \overline{T}_{ex} = \{p\}$ for all $p \in P$. However, by [9, Lemma 8] (a combinatorial argument) there is at least one placement p such that $|P \cap p + \overline{T}_{ex}| \geq \frac{1}{n}|\overline{T}_{ex}||P| \geq 2$ – a contradiction. □

In order to eliminate the remaining placements the robot visits the closest point $\overline{q}$ such that at least one placement is eliminated. Since $d_{\overline{T}}(\mathbf{0}, \overline{q})$ is no more than the shortest distance to localize the robot, a repeated application of this step guarantees that Step 4 has a competitive ratio of at most $\sqrt{n}$.

4 Searching for a Target in a Tree

As was pointed out in the introduction we are also interested in the searching variant of the problem where a robot has to find a target t marked on the map of the environment, but the robot is not given its starting position. We also apply Strategy LPS except with a slightly altered Step 4. Note that after Step 1 if t is not reached, then the distance between the starting position s of the robot and t is at least ϱ. Hence, after Step 3 the robot has traveled a distance of $O(\sqrt{n}d_T(s,t))$ and there are $k \leq \sqrt{n}$ possible placements for the robot and thus k possible locations $\overline{t}_1, \ldots, \overline{t}_k$ of t in $\overline{T}$. In Step 4 the robot now repeatedly attempts to visit the closest possible target among $\overline{t}_1, \ldots, \overline{t}_k$ until the true location of the target is identified. Obviously, the robot travels at most a distance of $\sqrt{n}d_T(s,t)$ in Step 4. This proves the following theorem.

Theorem 2 *Let T be a geometric tree and t be a point in T. There is a strategy that achieves a competitive ratio of $O(\sqrt{n})$ for a robot to find t given the coordinates of t in T.*

5 Conclusions

We have presented a new localization strategy for an autonomous mobile robot. The environment of the robot is represented by a geometric tree of constant degree in arbitrary dimensions. We assume that the robot knows its current orientation, and it has no use of vision other than to be able to detect the orientation of all edges incident to its current location. Our strategy achieves a competitive ratio of $O(\sqrt{n})$ if the tree contains n nodes of degree greater than or equal to three. Since there is a geometric tree which provides a lower bound example of $\Omega(\sqrt{n})$ for the competitive ratio of any localization strategy, our strategy is optimal up to a constant factor.

Challenges that remain for future work are to transfer the strategies developed for geometric trees to the more realistic setting of polygons in the plane and to investigate the complexity of the localization problem in graph structures that allow cycles.

References

[1] R. Baeza-Yates, J. C. Culberson, and G. J. E. Rawlins. Searching in the plane. *Information and Computation*, 106:234–252, 1993.

[2] E. Bar-Eli, P. Berman, A. Fiat, and P. Yan. Online navigation in a room. *J. Algorithms*, 17:319–341, 1994.

[3] M. Betke, R. Rivest, and M. Singh. Piecemeal learning of an unknown environment. In *Sixth ACM Conf. on Comput. Learning Theory*, pp. 277–286, July 1993.

[4] A. Blum, P. Raghavan, and B. Schieber. Navigating in unfamiliar geometric terrain. In *Proc. 23rd ACM Symp. Theory Comput.*, pp. 494–503, 1991.

[5] A. Blum and P. Chalasani. An on-line algorithm for improving performance in navigation. In *Proc. 34th IEEE Symp. Found. Comput. Sci.*, pp. 2–11, 1993.

[6] G. Dudek, K. Romanik, and S. Whitesides. Localizing a robot with minimum travel. In *Proc. 6th ACM-SIAM Symp. on Discrete Algos.*, pp. 437–446, 1995.

[7] L. Guibas, R. Motwani, and P. Raghavan. The robot localization problem in two dimensions. In *Proc. 3rd ACM-SIAM Symp. on Discrete Algos.*, pp. 259–268, 1992.

[8] B. Kalyanasundaram and K. Pruhs. A competitive analysis of nearest neighbor based algorithms for searching unknown scenes. In *Proc. 9th Symp. Theoret. Aspects Comput. Sci.*, pp. 147–157, 1992.

[9] J. M. Kleinberg. The localization problem for mobile robots. In *Proc. 35th IEEE Symp. Found. Comput. Sci.*, pp. 521–531, 1994.

[10] D. Miller, D. Atkinson, B. Wilcox, and A. Mishkin. Autonomous navigation and control of a Mars rover. In T. Nishimura, editor, *Automatic Control in Aerospace*, pp. 111–114, Pergamon Press, 1989.

[11] C. H. Papadimitriou and M. Yannakakis. Shortest paths without a map. In *Proc. 16th Internat. Colloq. Automata Lang. Program.*, pp. 610–620, 1989.

[12] C. Shen and G. Nagy. Autonomous navigation to provide long-distance surface traverses for Mars rover sample return missions. In *Proc. IEEE Intl. Symp. Intelligent Control*, pp. 362–367, 1989.

[13] D. D. Sleator and R. E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28:202–208, 1985.

[14] W. Thompson, H. Pick, B. Bennett, M. Heinrichs, S. Savitt, and K. Smith. Map-based localization: the 'drop-off' problem. In *Proc. DARPA Image Understanding Workshop*, pp. 706–719, 1990.

[15] Y. Yacoob and L. Davis. Computational ground and airborne localization over rough terrain. Technical Report CS-TR-2788, Univ. of Maryland, 1988.