Searching for Radio Beacons with Mobile Agents that Perceive Discrete Signal Intensities

Henning Hasemann*† Tom Kamphans*‡ Alexander Kröller*

Abstract

In this work, we present a number of algorithms for an agent to find a radio source in an unknown two-dimensional environment. We consider a model in which a mobile agent can perceive the intensity of a radio beacon only in discrete intervals instead of a uniform signal distribution.

Our solution for the general case also allows multiple local intensity maxima. Moreover, we present algorithms that behave efficiently if the width of intensity layers is bounded. For the case without geometric obstacles we also provide a lower bound for online algorithms and show that our solution operates within a constant factor of this bound.

1 Introduction

The problem of finding a radio source with mobile agents among geometric obstacles in an unknown environment has been studied intensively. The nature of the solutions vary substantially with the sensor model that is chosen for the implementing agent. A number of approaches focuses on a model in which the agent knows its exact position relative to the source at all times, such as the well-known *bug* family of algorithms [8]. Some of these approaches also grant the agent the ability to perceive parts of its environment directly [6, 3, 7]. Others require knowledge of only the direction of or the distance to the radio source [9, 2].

All of these approaches have in common that they use rather abstract models of radio-signal distributions or presume an information source other than a radio signal. However, real radio signals often do not provide direct angular or distance-related information, but still give enough information to find a direct path to the signal source [3, 11]. The idea is that even non-isotropic intensity landscapes may provide a path to the source, which can be found by following the direction of steepest signal intensity ascent.

Some real-world signal-intensity measurement systems, such as link-quality indicators (LQI, [5]), which are commonly used in wireless sensor networks, provide only discrete signal intensities. In such a setup, a signal intensity ascent may not be measurable without extensive motion. Thus, the approaches mentioned above cannot be applied. Also, there may be multiple (local) maxima of signal intensity at locations other than the radio source, due to reflection effects [10]. By combining covering techniques with search strategies, we are able to present algorithms that deal with some of these constraints and still guarantee to find the signal source.

In Section 2, we give some basic definitions that provide the groundwork for the following sections. In Section 3, we provide an approach that imposes only few constraints on the environments and still guarantees that an agent will eventually reach the signal source. Section 4 provides a lower bound for online algorithms in so-called *nested environments* without geometric obstacles and introduces an algorithm that operates within a constant factor of this bound. In Section 5, we present an algorithm that provides a maximum–path-length guarantee for the case with geometric obstacles. For a more comprehensive description, see Hasemann [4].

2 Preliminaries

An *intensity environment* (or *environment* for short) is given by a triple $E = (\mathcal{C}, p_{\text{target}}, \iota)$: the finite *configuration space* $\mathcal{C} = \mathcal{C}_{\text{free}} \dot{\cup} \mathcal{C}_{\text{blocked}} \subseteq \mathbb{R}^2$, the position of the signal source $p_{\text{target}} \in \mathcal{C}_{\text{free}}$ and an *intensity function* $\iota : \mathcal{C} \to \{1, \ldots, n_\iota\}$ with $\iota(p_{\text{target}}) = n_\iota$ where, $n_\iota \in \mathbb{N}$ is the number of different intensities that can occur in the environment. $\mathcal{C}_{\text{blocked}}$ consists of n_O obstacles $O_1, \ldots, O_{n_O}$ bounded by Jordan curves.

The set of *intensity layers*, $\mathcal{L}$, is a disjoint dissection of $\mathcal{C}$ into regions of constant value of ι. We require that every layer, $L_i \in \mathcal{L}$, is bounded by a set of Jordan curves and $\forall j \neq i : \partial L_i \cap \partial L_j \neq \emptyset \Rightarrow j \in \{i+1, i-1\}$. Regions that have no neighbor with higher ι-value are called *hills*. We call E a *nested environment* if every layer is connected, has one hole (except the layer containing p_{target}), and every layer with higher ι-value is fully contained inside this hole. Further, there is a *layer-width bound* $d \in \mathbb{R}$, such that for any $L \in \mathcal{L}$ with a hole $\forall p_l \in L : \exists p_h \in \mathcal{C} : (\iota(p_h) > \iota(p_l)) \wedge (||p_l - p_h|| \leq d)$ holds and the layer that contains p_{target} is contained in a circle with radius d around p_{target}.

For simplicity, we assume a point-shaped agent with

*Algorithms Group, Braunschweig Institute of Technology, Germany. http://www.ibr.cs.tu-bs.de/alg

†Supp. by 7th Framework Progr. contr. 258885 (SPITFIRE)

‡Supp. by 7th Framework Progr. contr. 215270 (FRONTS)

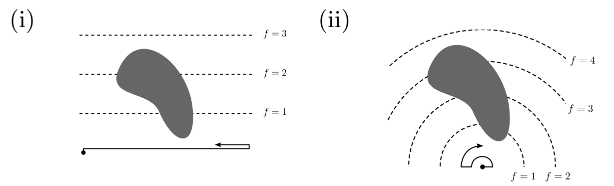

Figure 1: (i) Linear (lawn-mower–like) and (ii) circular covering patterns. Where the covering-motion curve (dotted lines) 'touches' an obstacle or an area of higher intensity, the configuration space is split into cells that are covered separately.

the capability to measure all signal intensities and to detect walls and the signal source in a radius r around its position. We assume that the agent has knowledge about whether or not the environment is nested and its layer width bound d (if any). However, the agent initially has no knowledge of the configuration space, the position of the signal source, the number of intensity layers, or the maximum intensity.

3 General Environments

We now present an algorithm that guarantees to find the signal source in general environments, using a simple back-and-forth covering strategy (e.g., sweep from left to right and back; see Fig. 1) as parameter. Note that such a strategy can cover only certain simply-connected areas, but in general is not sufficient to cover an unknown environment with obstacles and differing signal intensities. The covering pattern has to fulfill some criteria in order to make it usable for our algorithm and avoid certain corner cases. See the thesis on this subject [4] and the covering algorithm for environments without intensities [1] for details.

Algorithm 1: Intensity-Aided Coverage Planning (IACoP)

The agent starts in the outer layer (i.e., $\iota($ current position $) = 1)$.
Cover the current region using the given pattern, until an obstacle or an area of higher intensity is found. In this case, add the newly-found areas to a list. From this list, choose one of the areas with highest intensity and proceed with this area. If an area is completely covered, remove it from the list. Stop when p_{target} is found.

It is easy to see that the IACoP strategy finds the signal source, as it eventually covers $\mathcal{C}_{\text{free}}$ completely. Although this algorithm has the upside of being very general in terms of constraints imposed on the environment, it can be forced to cover an area that is arbitrarily large in comparison to the agent's distance to the source. In the following section, we will see that nested environments can be searched more efficiently.

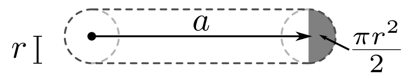

Figure 2: $A_{\text{tube}}(a)$

4 Nested Environments without Obstacles

In this section, we consider nested environments with a layer width bound d. We provide a lower bound on the agent's path length, and an algorithm for the case $\mathcal{C}_{\text{blocked}} = \emptyset$; the case with obstacles is discussed in the following section.

4.1 Lower Bound

Lemma 1 *Let $A \subset \mathbb{R}^2$ be an area with $|A| < \infty$. A path of length $l_{\text{covmin}}(|A|) = \min\{(|A| - \pi r^2)/2r, 0\}$ is always necessary and sometimes sufficient for an agent with coverage radius r to cover A completely.*

Proof. Assume that there is an area A with $\pi r^2 \leq |A| < \infty$ that is coverable by a path of length $l < l_{\text{covmin}}(|A|) = \frac{|A|-\pi r^2}{2r}$. Initially (i.e., prior to any actual motion), the agent covers $|A_{\text{init}}| \leq \pi r^2$. Then, during each motion step of length ε, the agent covers an additional area A_ε with $|A_\varepsilon| = \frac{|A|-|A_{\text{init}}|}{l}\varepsilon$ on average. We observe that $|A_\varepsilon| > 2r\varepsilon\frac{|A|-|A_{\text{init}}|}{|A|-\pi r^2} \geq 2r\varepsilon$ holds. However, for a step of length ε, the agent cannot do better than moving on a straight line into uncovered space. Such a step cannot cover more than $2r\varepsilon$ which is a contradiction to $|A_\varepsilon| > 2r\varepsilon$. Thus, an agent cannot cover any area A with a path shorter than $l_{\text{covmin}}(|A|)$.

The area $A_{\text{tube}}(a)$, see Fig. 2, with $|A_{\text{tube}}(a)| = 2ra+\pi r^2$ can clearly be covered by an agent moving on a straight line of length $a = (2ra + \pi r^2 - \pi r^2)/2r = l_{\text{covmin}}(|A_{\text{tube}}(a)|)$. Thus, a path of length $l_{\text{covmin}}(|A|)$ is sometimes sufficient for covering an area A. $\square$

Theorem 2 *For any online algorithm* ONL *that finds the signal source in a nested environment without obstacles, there exists an environment E with layer width bound $d \geq 2r$ and $n_\iota \geq 3$ intensity layers, such that an agent with a coverage radius of r executing ONL in E travels a path of length $D_{\text{ONL}} \geq (n_\iota - \frac{1}{2})d + \frac{\pi d^2}{2r} + (\frac{1}{2} - \frac{3\pi}{4})r$.*

Proof. We construct E, such that an agent must walk a path of length $(n_\iota - 1)d$ in order to reach the hill. Let p_{n_ι} be the first point of the hill that is covered by the agent. We place the signal source at the point in the disk of radius d around p_{n_ι} that is last to be covered by the agent. Note that the agent must already have covered (slightly less than)

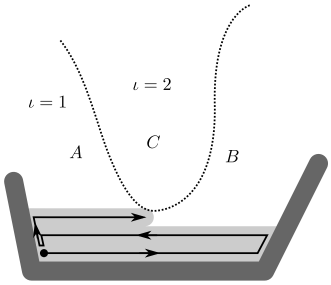

Figure 3: A mobile agent covering the environment. When detecting an area of different intensity, the agent creates three new cells A, B and C. The one with the highest intensity (C) will be covered first.

$|A_{\text{tube}}(d-r)| - \frac{\pi}{2}r^2$ of that disk, so the length of the path an agent travels after it has detected p_{n_ι} in E is at least $l_{\text{covmin}}(|A_{\text{tube}}(d-r)| - \frac{\pi}{2}r^2)$.

Altogether, the agent has to travel a distance of at least $(n_\iota - \frac{1}{2})d + \frac{\pi d^2}{2r} + (\frac{1}{2} - \frac{3\pi}{4})r$ in order to reach the signal source. $\square$

We conjecture that it is possible to construct a better lower bound by exploiting the fact that the mobile agent cannot know in which direction an area of higher intensity is located. However, it is difficult to predict a minimal path length, because an optimal path strongly depends on the shape of the areas that the agent has already covered.

4.2 Circular Motion Planning

Algorithm 2: Circular Motion Planning (CMoP)

c_1 := current position;
$i := 1$;
walk an arbitrary straight line of length $d - r$;
repeat
  walk on a circle around c_i;
  on intensity increase detected at some point p
    $i := i + 1$;
    $c_i := p$;
    go to nearest point q with $\|c_i - q\| = d - r$;
until *circle complete*;
move on concentric circles towards c_i;

Theorem 3 *An agent executing the CMoP algorithm in a nested environment will reach the radio source with a total maximum travel distance of* $D_{\text{CMoP}} \leq 8.192\, D_{\text{ONL}}$.

Proof. When the agent starts executing the algorithm, it travels a straight line of length $d - r$, followed by at most a complete circle of length $2\pi(d-r)$ around c_1.

For the following circles around a point c_k, $k \in \{2, \ldots, n_\iota - 1\}$ a point p of intensity $\iota(p) > \iota(c_k)$ can either be found during circling or not. If it is found, a new circle is started around $p = c_{k+1}$ and the way costs for the circle around c_k are $d - 2r$ for getting to the closest point on the circle and at most $2\pi(d-r)$ for the actual circular motion.

If the agent has not found a point with higher signal intensity than $\iota(c_k)$ so far, such a point must be inside the d-disk around c_k. As the environment is nested, the same holds true for all other points with higher intensity than $\iota(c_k)$, especially the radio source.

In the last step, we walk on $\lceil \frac{d}{2r} \rceil$ concentric circles towards c_k and need a total of d for the relocation between two successive cirlces. From the last circle, we move a distance of r towards the source:

$$\begin{aligned} D_{\text{CMoP}} &\leq (2\pi{+}1)(d{-}r){+}((2\pi{+}1)d{-}(2\pi{+}2)r)(n_\iota{-}2) \\ &\quad + \textstyle\sum_{i=1}^{\lceil \frac{d}{2r} \rceil} 2\pi r(2i-1) + d + r \\ &\leq (2\pi+1)\,(n_\iota-2)\,(d-r) + \tfrac{\pi d^2}{2r} + 4\pi d + 2 \\ &< (2\pi+1) \cdot \tfrac{n_\iota}{n_\iota - \frac{1}{4} - \frac{3}{8}\pi} \cdot D_{\text{ONL}} \\ &\leq 8.192\, D_{\text{ONL}} \end{aligned}$$ $\square$

We observe that for high values of n_ι, this bound approaches $2\pi + 1$ (≈ 7.2832).

5 Nested Environments with Obstacles

For nested environments with obstacles, we use the Circular Coverage Planning Algorithm (CCoP), see Alg. 3. Figure 4 shows an example of this algorithm.

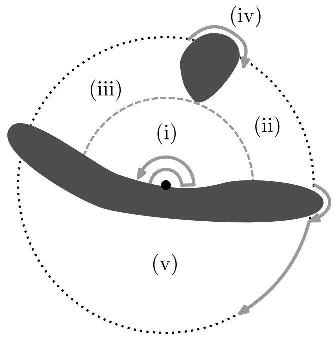

Figure 4: Example run of the CCoP algorithm. The agent starts covering area (i) with a circular pattern until it detects an obstacle (dashed grey line). Next, areas (ii) and (iii) are covered separately. Then the agent follows the nearest obstacle wall (iv) and marks the obstacle as covered. The agent follows another obstacle wall and covers the newly found part of the d-disk (v) starting with large circles.

Theorem 4 *An agent executing the CCoP algorithm will reach the radio source with a total maximum*

Algorithm 3: Circular Coverage Planning Algorithm (CCoP)

Let p be the agent's starting position.
- Walk a concentric circle pattern around p.
- Whenever a wall piece is detected, store its position in a list. Also, store the positions of the sub-areas to both sides of the wall piece.
- When the current circle cannot be continued (either because it is complete or because of an obstacle), recursively cover all known but not covered cells by extending the circular pattern to those areas.
- When the current circle radius has reached d, no point of higher intensity can be found, and all known cells are covered, follow the nearest wall piece until an area of higher intensity or an uncovered part of the d-circle around p is found. In the latter case, cover this part, then continue following the wall so that it will eventually be circled completely.
- Whenever an obstacle has been circled completely, mark the area of the obstacle as already covered.
- Whenever a point p' with intensity higher than $\iota(p)$ is detected, move to p and start the algorithm from the beginning with $p := p'$.
- When the signal source is detected, move to the source, and stop.

travel distance of
$D_{\text{CCoP}} \leq (\frac{\pi d^2}{2r}+(2\pi+1)d+(2\pi+2)r)n_\iota+2\sum_{i=1}^{n_O}|\partial O_i|$.

Proof. The agent starts with a covering loop around its starting position. The actual covering happens in $\lceil \frac{d}{2r} \rceil$ circles with radii $r, 3r, 5r, \ldots \left(2\lceil \frac{d}{2r} \rceil - 1\right) r$. Then the circular part of the covering loops is bounded by $\sum_{i=1}^{\lceil \frac{d}{2r} \rceil} ((2i-1)2\pi r)$.

For relocation between covering loops, the agent walks a distance of at most d when there are no obstacles. In addition, the agent will walk n_ι path segments, each of length no more than r, for getting from one spiral to the next or to the source.

When there are obstacles that intersect the covering loop, the relocations happen on the obstacle boundaries, and thus are bounded by $\sum_{i=1}^{n_O}|\partial O_i|$. The search for an unexplored part of the covering circle is also a path along obstacle borders, so its length is bounded by $\sum_{i=1}^{n_O}|\partial O_i|$, too.

In total we get

$$\begin{aligned} D_{\text{CCoP}} &\leq \left[\sum_{i=1}^{\lceil \frac{d}{2r} \rceil} ((2i-1)2\pi r) + 2\left\lceil \frac{d}{2r} \right\rceil r\right] n_\iota \\ &\quad + 2\sum_{i=1}^{n_O}|\partial O_i| \\ &\leq \left[(2\pi+1)d + \frac{\pi}{2}\frac{d^2}{r} + (2\pi+2)r\right] n_\iota \\ &\quad + 2\sum_{i=1}^{n_O}|\partial O_i| \end{aligned}$$

□

6 Conclusion

We presented approaches for three different cases of environments with discrete signal intensities. In the case of environments with a layer width bound and no obstacles we could show that our algorithm is within a constant factor of an optimal online algorithm.

Acknowledgements

We thank Sándor P. Fekete for helpful discussions and proofreading.

References

[1] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull. Morse decompositions for coverage tasks. *Internat. J. Robot. Res.*, 21(4):331–344, 2002.

[2] D. Angluin, J. Westbrook, and W. Zhu. Robot navigation with distance queries. *SIAM J. Comput.*, 30:110–144, 2000.

[3] M. Burger, Y. Landa, N. Tanushev, and R. Tsai. Discovering point sources in unknown environments. In *Proc. Workshop Algorithmic Found. Robot.*, 2008.

[4] H. Hasemann. Searching for radio beacons among geometric obstacles with mobile agents. Master's thesis, Braunschweig Institute of Technology, 2010. http://www.ibr.cs.tu-bs.de/users/hasemann/.

[5] IEEE. Standard 802.15.4 for local and metropolitan area networks specific requirements part 15.4, 2006.

[6] I. Kamon and E. Rivlin. Sensory-based motion planning with global proofs. *IEEE Trans. Robot. Autom.*, 13:814–822, 1997.

[7] V. J. Lumelsky and T. Skewis. A paradigm for incorporating vision in the robot navigation function. In *Proc. IEEE Internat. Conf. Robot. Automat.*, pages 734–739, 1988.

[8] V. J. Lumelsky and A. A. Stepanov. Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica*, 2:403–430, 1987.

[9] V. J. Lumelsky and S. Tiwari. An algorithm for maze searching with azimuth input. In *Proc. IEEE Internat. Conf. Robot. Automat.*, pages 111–116, 1994.

[10] L. Tang, K.-C. Wang, Y. Huang, and F. Gu. Channel characterization and link quality assessment of ieee 802.15.4-compliant radio for factory environments. *IEEE Trans. Industrial Informatics*, 3(2):99–110, 2007.

[11] K. Taylor and S. M. LaValle. I-bug: An intensity-based bug algorithm. In *Proc. IEEE Internat. Conf. Robot. Automat.*, pages 3981–3986, 2009.