

# Towards Distance Ordering in Large Robot and Sensor Networks

Megha Gupta and Gaurav Sukhatme

July 19, 2010

## Abstract

Localization in robot networks has been extensively studied and in the absence of sophisticated sensors and high connectivity, it remains a challenging problem. We study the problem of distance ordering i.e. comparing any two distances in a large robot network and deciding which one is larger. Given a localized graph, distance ordering is trivial but the question we attempt to answer is whether it can be done without localization. We first present a survey of results on the localizability of a network given a variety of inputs and infer that it is an NP-hard problem except in certain special scenarios. We then present some preliminary results on tackling distance ordering without localization.

## 1 Introduction

Localization in robot networks is an extremely well-researched topic. Each robot in the network can have a variety of sensors including laser range finders, compass, and radio antennae. Presence of a few anchor nodes that are fixed and know their locations is often assumed. In spite of an extensive literature on the subject, localization remains a challenging problem in the absence of sophisticated sensors and high connectivity. And in a large robot network, both the use of sophisticated sensors and high connectivity cannot be assumed.

We, therefore, attempt to answer the question that whether distance ordering, i.e., the comparison of any two inter-robot distances in a large network be done reliably without finding the exact coordinates of the nodes. For applications like maintaining connectivity and routing, it would often suffice to know if distance  $x$  is larger than distance  $y$  without precisely knowing either  $x$  or  $y$ . Given a localized graph, distance ordering is, of course, trivial but can it be done without localization?

We first present a survey of results on the localizability of a network given a variety of sensor inputs. We then present some preliminary results on solving the distance ordering problem without localization.

## 2 Localizability of Networks

Every node in a robot network may have a variety of sensors that enable it to find its neighbors and measure distances and angles between them. We present a short survey of results on the localizability of a robot network given various combinations of these inputs without any anchor nodes. It is obvious that distance ordering is trivial once a network is localized.

1. *Input:* Only connectivity graph

**Theorem 2.1.** *Localization is NP-hard for unit disk graphs [2].*

With only a bound on the inter-neighbor distance known, there isn't much that can be done about distance ordering either.

2. *Input:* Connectivity graph, distances between neighbors

**Theorem 2.2.** *Provided the distances are accurate, localization, and hence distance ordering, can be done if the underlying grounded graph is generically globally rigid [1].*

However, it is NP-hard to find if a given framework is globally rigid [4]. It is NP-hard even to test whether a unit-disk framework is globally rigid [1].

3. *Input:* Connectivity graph, angle magnitudes between two consecutive neighbors of a robot, neighbor ordering

**Theorem 2.3.** *Localization is NP-hard for unit disk graphs [3].*

Bruck et al. [3] also found a technique to construct a planar spanner subgraph for this case with a spanning ratio of 2.42.

4. *Input:* Connectivity graph, angle magnitudes between two consecutive neighbors of a robot

**Lemma 2.4.** *Localization is NP-hard for this case.*

*Proof.* Let us call an instance of the unit-disk localization problem given the connectivity graph, angle magnitudes, and neighbor ordering as  $\mathcal{CAN}$  and an instance given only the connectivity graph and angle magnitudes as  $\mathcal{CA}$ . From Theorem 2.3, we know that  $\mathcal{CAN}$  is NP-hard.

Now let us assume that an easy (polynomial time) algorithm  $A$  exists that can solve  $\mathcal{CA}$ . Then, using  $A$  to solve  $\mathcal{CAN}$ , we can simply throw away the redundant input of neighbor ordering so that we are left with an instance of  $\mathcal{CA}$  and solve  $\mathcal{CAN}$  in polynomial time.

This is a contradiction to Theorem 2.3.

Therefore,  $\mathcal{CA}$  is NP-hard. □

5. *Input:* Connectivity graph, neighbor ordering

**Lemma 2.5.** *Localization is NP-hard for this case.*

*Proof.* Let us call an instance of the unit-disk localization problem given the connectivity graph, angle magnitudes, and neighbor ordering as  $\mathcal{CAN}$  and an instance given only the connectivity graph and neighbor ordering as  $\mathcal{CN}$ . From Theorem 2.3, we know that  $\mathcal{CAN}$  is NP-hard.

Now let us assume that an easy (polynomial time) algorithm  $A$  exists that can solve  $\mathcal{CN}$ . Then, using  $A$  to solve  $\mathcal{CAN}$ , we can simply throw away the redundant input of neighbor ordering so that we are left with an instance of  $\mathcal{CN}$  and solve  $\mathcal{CAN}$  in polynomial time.

This is a contradiction to Theorem 2.3.

Therefore,  $\mathcal{CN}$  is NP-hard. □

Note that neighbor ordering can be deduced from angle magnitudes and angle signs together and angle signs can be deduced from neighbor ordering and angle magnitudes together. However, angle magnitudes cannot be deduced from neighbor ordering and angle signs.

6. *Input:* Connectivity graph, distances between neighbors, neighbor ordering

Neighbor ordering implies the order in which a robot sees its neighbors when it rotates at its position in a given direction, say counter-clockwise. If a robot  $A$  has only two neighbors,  $B$ , and  $C$ , neighbor ordering is meaningless since with the same positions of the 3 robots,  $A$  can see either  $B \rightarrow C$  or  $C \rightarrow B$  depending on its initial orientation (Fig. 1). However, with three neighbors, flip ambiguities in robot locations can be resolved as shown in Fig. 2.

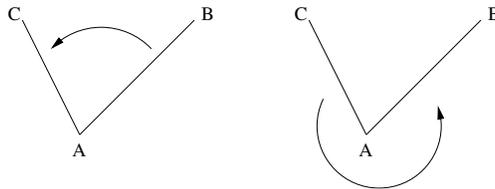


Figure 1: Neighbor ordering is meaningless with only two neighbors.

Let  $K_3$  denote a complete graph on 3 vertices. Starting with a  $K_3$  with all three vertex positions known, we can localize any new point that is a common neighbor to at least two known points. To understand this better, see Fig. 3. Triangle  $ABC$  is a  $K_3$  and locations of  $A$ ,  $B$ , and  $C$  are accurately known. Say  $D$  is a common neighbor to both  $A$  and  $B$  and its distance to each of them is known. Using the known locations of  $A$  and  $B$ , we can find at most two possible locations of  $D$ , shown by  $D$  and  $D'$  in the figure. Now using neighbor ordering,  $A$  can check if the order is  $B - D - C$  or  $B - C - D$ . Similarly,  $B$  can also check its neighbor ordering and know whether the correct location of  $D$  is at  $D$  or  $D'$ .

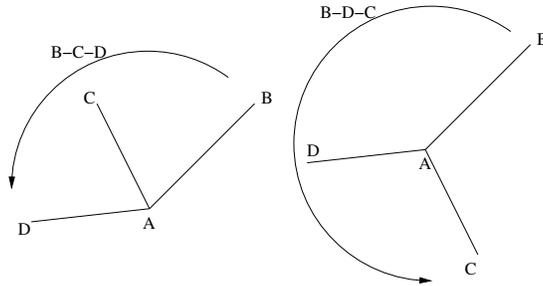


Figure 2: Neighbor ordering helps resolve flip ambiguities when a robot has at least three neighbors.

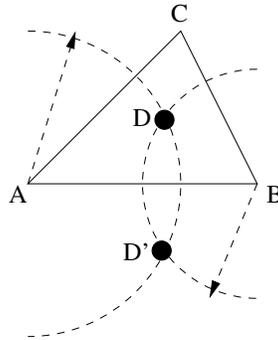


Figure 3:

Notice that  $D$  forms another  $K_3$  with  $A$  and  $B$  and thus, a graph that can be localized by this technique will have at least two  $K_3$ 's out of which one needs to be completely known in the beginning. Also note that each node in this graph has at least two neighbors and all inter-neighbor distances are known. Thus, a point is restricted to at most two possible locations, the correct one of which can be decided by using neighbor ordering. Thus, such a graph is necessarily rigid. The knowledge of neighbor ordering helps us resolve flip ambiguities and thus, relaxes the constraint of a graph to be globally rigid to be localizable.

7. *Input:* Connectivity graph, distances between neighbors, angle magnitudes between two consecutive neighbors of a robot

As in the case of neighbor ordering, knowledge of angles between neighbors does not help when a robot has only two neighbors as shown in Fig. 4. However, when a robot has at least three neighbors, flip ambiguities can be resolved as shown in Fig. 5.

As before, starting with a  $K_3$  with all three vertex positions known, we can localize any new point that is a common neighbor to at least two known points and resolve flip ambiguities using the angle magnitude information. Thus, the knowledge of angle magnitudes also relaxes the constraint of global rigidity for a graph to be localizable.

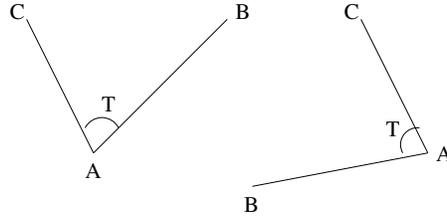


Figure 4: Knowledge of angles between neighbors does not help when a robot has only two neighbors.

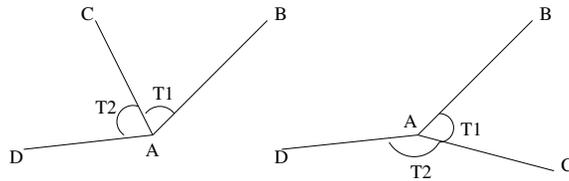


Figure 5: Flip ambiguities can be resolved using angle magnitudes when at least three neighbors are present.

8. *Input:* Connectivity graph, angle magnitudes between two consecutive neighbors of a robot, neighbor ordering, distances between neighbors

This gives complete information about the underlying graph. Hence localization and distance ordering become trivial in this case.

### 3 Distance Ordering Without Localization

Framework: We know the connectivity graph and some combination of inputs (measurements) that is not sufficient to localize the network. Can we do distance ordering? How well can we do?

If all inter-neighbor distances are known, comparing the lengths of any two edges is trivial and needs no localization. Consider the case when distances are not known. The smallest polygon possible in a graph is the triangle and for a triangle, congruence is determined by *SSS*, *ASA*, *SAS*, and *AAS* where *S* denotes an edge length and *A* denotes an angle contained between two edges. Congruence is not determined by *AAA* and *SSA*. The nodes of a triangle can be localized relative to each other if congruence is determined. So consider the case *AAA* where localization is not possible but distance ordering is because the lengths of the sides are related to the angles as (Fig. 6(a)):

$$\frac{\sin A}{a} = \frac{\sin B}{b} = \frac{\sin C}{c}$$

It can be proved that ratios of the sides of a quadrilateral or a higher degree polygon cannot be calculated even if all the angles are known provided that is the only information known. As shown in Fig. 6(b), length of side *AB* is kept fixed while *CD* is continuously

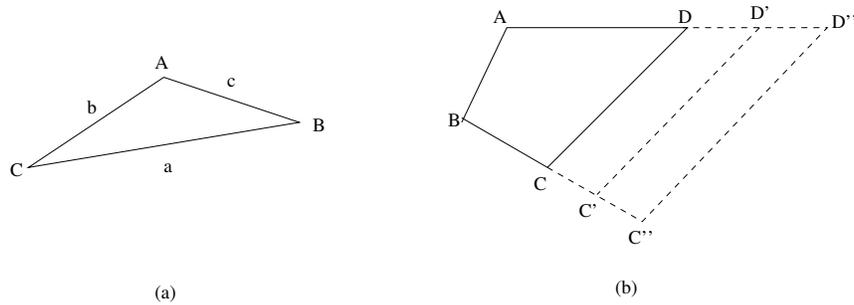


Figure 6: (a) The ratio of side lengths to the opposite angles in a triangle is constant. (b) Even if all angles of a quadrilateral are known, the ratio of its sides cannot be calculated. In this figure,  $AB$  is constant while  $CD$  continuously changes without changing any of the angles.

moved parallel to itself keeping all angles intact but resulting in arbitrary ratios of the sides of the quadrilateral  $ABCD$ .

So for a network where robots can only measure the angles between their neighbors, distance ordering without localization is possible only if the network graph is a *chordal* or *triangulated* graph i.e. a graph where at least one chord or an edge exists between two non-adjacent nodes in a loop. If a higher-degree polygon is present, any of its edges that

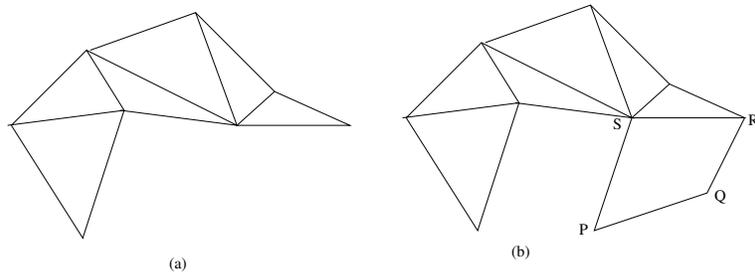


Figure 7: (a) If all the angles of this graph are known, the ratio of any two edges can be calculated and hence, distance ordering can be done. (b) Since the edge  $RS$  of the quadrilateral  $PQRS$  is adjacent to the triangulated component of the graph, its ratio to any edge of the triangulated component can be calculated.

are adjacent to a triangulated part of the graph can be compared to any other edge of the triangulated component (for example, see Fig. 7(b)).

This leads us to the following conclusion: any two edges of a graph can be exactly compared iff there is a single connected triangulated component of the graph that is adjacent to every non-triangulated component (Fig. 8).

It is important to note that for these special networks, if the length of any edge is known, we obtain all edge lengths exactly. With all distance and angles known, the network is localized. Thus, this solution is just an edge length away from exact localization.

Now consider the case when a robot can possibly order some of the edges incident on it

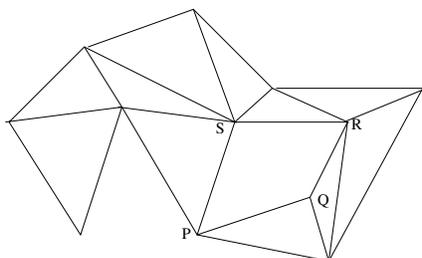


Figure 8: Any two edges of this graph can be exactly compared as it has a single connected triangulated component that is adjacent to every edge of the non-triangulated component.

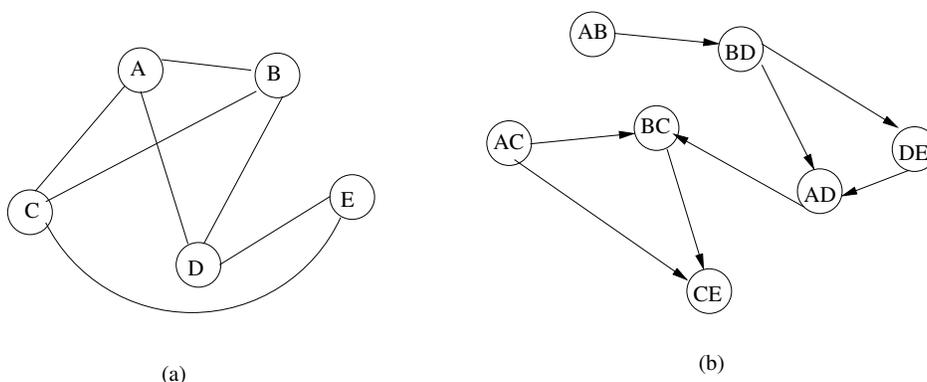


Figure 9: (a) The connectivity graph. (b) The corresponding edge-graph obtained by combining the local order information.

and this 'local order' of each robot is collected together. Thus we obtain a partial ordering of edges on a graph and our goal is to deduce the global edge ordering from it.

Let  $G = (V, E)$  be the given graph and  $A$  be the set of ordered pairs of edges (see Fig. 9(a)). Construct  $G' = (V', E')$  such that  $V' = A$  and for every ordered pair  $(u, v)$  in  $A$ , there exists a directed edge from  $u$  to  $v$  in  $E'$  as shown in Fig. 9(b).

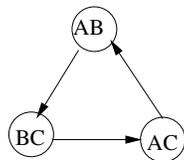


Figure 10: A cycle in the edge-graph indicates an inconsistency in distance ordering.

Finding a global ordering on the edges of  $G$  is equivalent to finding a Hamiltonian path in  $G'$  which is an NP-complete problem for generic graphs. However,  $G'$  is a directed acyclic graph (DAG) because cycles in  $G'$  would imply an inconsistency in the distance ordering. For example, in Fig. 10, the cycle implies  $AB < BC < AC < AB$ . For DAGs, dynamic

programming can be used to find a Hamiltonian path if it exists in polynomial time.

## 4 Conclusion

We studied the problem of ordering inter-robot distances in a large robot network. Our research indicates that this is a hard problem without having localization as an intermediate step and can be exactly solved only for a very special class of networks - triangulated networks. Given local orderings of edges, finding a global distance ordering translates to finding a Hamiltonian path in a directed acyclic graph (DAG). Polynomial time algorithms using dynamic programming exist that would find the Hamiltonian path in the DAG if it exists. Work in progress involves exploiting mobility of the robots to solve distance-ordering for arbitrary networks.

## References

- [1] James Aspnes, Tolga Eren, David K. Goldenberg, A. Stephen Morse, Walter Whiteley, Yang Richard Yang, Brian D. O. Anderson, and Peter N. Belhumeur. A Theory of Network Localization. *IEEE Transactions on Mobile Computing*, 5(12):1663–1678, 2006.
- [2] Heinz Breu and David G. Kirkpatrick. Unit Disk Graph Recognition is NP-Hard. *Computational Geometry. Theory and Applications*, 9, 1993.
- [3] Jehoshua Bruck, Jie Gao, and Anxiao (Andrew) Jiang. Localization and routing in sensor networks by local angle information. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 181–192, New York, NY, USA, 2005. ACM.
- [4] J.B. Saxe. Embeddability of weighted graphs in k-space is strongly NP-hard, 1979.