Distributed networked localization using neighboring distances only through a computational topology control approach

Xiaochu Wang\(^1\), Ting Sun\(^2\), Changhao Sun\(^1\) and Junqi Wang\(^3\)

Abstract
For large-scale wireless sensor networks, the nonlinear localization problem where only neighboring distances are available to each individual sensor node have been attracting great research attention. In general, distributed algorithms for this problem are likely to suffer from the failures that localizations are trapped in local minima. Focusing on this issue, this article considers a fully distributed algorithm by introducing a novel mechanism, where each individual node is allowed to computationally interact with a random subset of its neighbors, for helping localizations escape from local minima. Theoretical analyses reveal that with the proposed algorithm, any local minimum of the localization will be unstable, and the global optimum would finally be achieved with probability 1 after enough time of iterations. Numerical simulations are given as well to demonstrate the effectiveness of the algorithm.

Keywords
Node localization, distributed algorithm, global optimum, sensor networks, topology control

Introduction
Wireless sensor networks have been attracting great attention since a large number of simple sensor nodes working cooperatively can bring plenty of advantages. With the increasing requirement of cooperations among sensors, the problem of node localization arises since position information plays a crucial role in complex collaboration. However, as the network scale grows, manually configuring positions for each node would become a disaster, and equipping all nodes with high-precision Global Navigation Satellite System (GNSS) receivers not only costs much but also is strongly environment dependent. Nowadays, it is widely believed that localizing nodes using relative measurements is of great benefits to large-scale sensor networks. Among varieties of inter-node information, the relative distance is favorite due to its low-cost economy, unlimited field of view, and high precision. Therefore, plenty of research interest has been focused on the localization using relative distances. However, the main issue troubling the research is that even in the absence of measurement noises, the distance-based localization is a nonconvex optimization problem with many local minima.

Corresponding author:
Changhao Sun, Qian Xuesen Laboratory of Space Technology, China Academy of Space Technology, Beijing 100080, China.
Email: wangxiaochu@qxslab.cn

Creative Commons CC BY. This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
In general, node localization algorithms can be classified into two categories: centralized ones and distributed ones. The centralized ones mainly include elaborately designed heuristic optimization algorithms, such as the genetic algorithm, simulated annealing algorithm, ant colony optimization, and particle swarm optimization for localization. Besides, kinds of model-based nonlinear fitting algorithms are also included, where a central model of the relative structure with respect to node positions is used for collecting and fitting all the distance measurements among nodes. However, centralized algorithms completely rely on global information and centralized computations, put forward heavy requirements on network conditions, and lack of the feasibility and flexibility for network scales, despite the fact that they can obtain an excellent solution after running for enough time of iterations.

On the contrary, distributed algorithms behave in a decentralized manner without using any global information. Further divisions of distributed algorithms can be made, according to whether the localization process is simultaneous for all the nodes or not, into incremental algorithms and concurrent algorithms, respectively. The former algorithms locate nodes sequentially in an incremental way, where positions are solved by starting with the group of anchor nodes, and from the close-by nodes to those far away. In each iteration, any node whose position has been solved is designated as an anchor node. These algorithms are simple to understand and easy to implement. However, they are error-prone because errors would be accumulated along the solving paths. This drawback can cause a fatal localization crash for large-scale networks. In contrast, the latter algorithms solve positions for all the nodes simultaneously, where each individual node in the network updates its position estimate synchronously in each iteration following the same protocol. In general, concurrent algorithms are more competent for large-scale networks and become the research emphasis in distributed localization problems.

In the studies of concurrent algorithms, many works focus on the problem of localization refinement, where rough position estimates are desired to be refined to more accurate ones. Toward this goal, the position estimate of each node is often considered as a distribution rather than a single point, and nodes then refine their position estimates by modeling the distribution characteristics, by intersecting the neighboring distributions, and by fusing data within the intersections. If the models of both the localization process and the noises are available, then estimates can be refined by making the noisy measurements best fit the model from the perspective of probability distribution, and thus, a refined localization result with extremely high precision would be obtained. However, the refinement is deeply restricted by two major assumptions, without which the process would fail. First, the models are assumed to be exact enough to describe the actual signals. Second, rough position estimates are assumed to be known prior to nodes. For getting a proper starting point for the refinement localization, research turns to the study of the lost-in-space localization, where nodes have no prior knowledge on positions and need to produce position estimates from a large solution space. Toward this goal, position estimates can never be considered as distributions, since the initial error of starting points would probably go beyond the assumed bound of probability distribution models. Generally, most research treats the position estimate as a single point instead. In the studies by Priyantha et al., Howard et al., and Gotsman and Koren, a distributed localization algorithm, which can be referred to as the mass–spring relaxation (MSR) algorithm, is developed using the gradient descent method for guaranteeing the convergence to local optima. MSR regards the estimates on node positions as distinct virtual masses, and the resulting distances among masses as springs. Depending on the error between the resulting distance and the measured one, every spring is recognized to be either “compressed” or “stretched.” Along the gradient direction, a compressed spring provides a repulsive force that pushes two neighboring nodes away, and a stretched spring provides an attractive force that pulls the nodes near. Virtual masses will thus cooperatively keep moving under spring forces until they get a balanced assignment, which is regarded as the solution of localization. However, the balanced assignment using MSR only corresponds to local minima, because the distance-based localization optimization is of high nonconvexity even in the absence of measurement noises.

To solve the local-minima issue, varieties of methods have been considered. For example, the information of hop-counts among nodes can be used in advance to generate an excellent starting point that is close to the global minimum, for the followed localization process using MSR algorithm. But the starting point cannot be always good enough, especially when the network connectivity is large. The information of additional mobile anchor nodes is also helpful, because varying distances can be introduced to save the localization from local minima. But this method brings difficulties in engineering implementation as well. Without requiring any additional information, Zhu and Ding proposed an MSR-based “pulled-only” algorithm, where only the stretched springs are retained while the compressed ones are ignored. But this method requires that the problem be convex, that is, all ordinary nodes need to lie inside the convex hull of the anchor nodes. Naraghi-Pour and Rojas proposed an observer for monitoring localization errors and used an error-triggered strategy. Under the strategy, active
disturbance noises are additionally put on position estimates whenever the observed localization errors are large. This method is suitable for dealing with nonconvex problems, but it brings additional issues to designing toggle conditions and generation noises. Besides, the global optimum would be unstable under this method, since toggle conditions cannot be exact in the absence of uncertain measurement noises. We should point out that for the sake of brevity, our survey of previous work here is short of complete and only contains the previous work related to our work. For a more complete survey, we refer to the studies by Yassin et al., Paul and Sato, and Naraghi-Pour and Rojas.

Focusing on the local-minima issue in distance-based localization problems, this article considers a concise topology-controlled mass–spring relaxation (TCMSR) algorithm that allows node to computationally interact with only a random subset of its neighbors, for help localizations escape from local minima. The significant contributions of the algorithm are fourfold: (1) the global minimum would still be stable under the algorithm, while any local minimum would not; (2) the algorithm would finally achieve the global optimum with probability 1 after enough iterations; (3) no additional information is needed for assisting the localization. Actually, on the contrary, less information is used in the algorithm; (4) the algorithm addresses the issue in the topology control level only and does not conflict with other existing algorithms, meaning that TCMSR can be used for improving these existing algorithms.

The rest of this article is organized as follows: Section “Preliminaries” reviews the preliminaries of the graph and the traditional mass–spring algorithm. Section “TCMSR algorithm” formally proposes the TCMSR algorithm along with theoretical analyses and discussions. In section “Simulations,” numerical simulations are performed and show the effectiveness of the proposed algorithm. Finally, remarking conclusions are given in section “Conclusion.”

**Preliminaries**

**Notation and definition**

Consider a network consisted of $N$ ordinary nodes and $M$ anchor nodes at distinct positions in some physical region, where the anchor node can be aware of its own position alone while the ordinary node cannot. By labeling the ordinary nodes by $1, 2, \ldots, N$ and the anchor nodes by $N + 1, N + 2, \ldots, N + M$, we denote the ordinary node set by $\mathcal{N} = \{1, 2, \ldots, N\}$ and the anchor node set by $\mathcal{M} = \{N + 1, N + 2, \ldots, N + M\}$. The interaction relationship among the nodes can be modeled by a directed graph $\mathcal{G}$. A directed edge $e_{ij}$ in $\mathcal{G}$ means that node $i$ can receive information from node $j$, but not necessarily vice versa. To indicate the existences of the edges, the adjacency matrix $A = [a_{ij}]$ associated with $\mathcal{G}$ is defined such that $a_{ij} = 1$ if the edge $e_{ij}$ exists and $a_{ij} = 0$ otherwise. The Laplacian matrix $L = [l_{ij}]$ associated with $\mathcal{G}$ is also defined such that $l_{ii} = \sum_{j \in \mathcal{N}} \cup_{\mathcal{M}, i \neq j} a_{ij}$ while $l_{ij} = -a_{ij}, \forall i \neq j$. We say node $j$ is a neighbor of node $i$ if $a_{ij} = 1$ and denote the neighbor set of node $i$ by $\mathcal{N}_i \subset \mathcal{N} \cup \mathcal{M}$ such that $a_{ij} = 1, \forall j \in \mathcal{N}_i$. Therefore, $|\mathcal{N}_i|$ denotes the number of neighbors of node $i$. The neighbor relationship $j \in \mathcal{N}_i$ does not imply $i \in \mathcal{N}_j$ in a directed graph, but note that $A$ is symmetric in undirected graphs. A graph is said to contain a directed spanning tree if there is a directed path that starts from an anchor node in the graph and connects all the other ordinary nodes. As an example, Figure 1 illustrates the sketch of a directed sensor network that contains directed spanning trees.

Denote by $r^*_i$ the actual position of each node $i \in \mathcal{N} \cup \mathcal{M}$ with respect to a certain coordinate system, and suppose some mechanism exists through which ordinary nodes can measure the distances to its neighbors. Denote the directed distance measurements by $d_{ij}$ for each ordinary node $i \in \mathcal{N}$ and its neighbors $j \in \mathcal{N}_i$, such that $d_{ij} = |r^*_i - r^*_j| + \epsilon_{ij}$, with $\epsilon_{ij}$ being a small bounded measuring noise.

The localization problem can be then defined in the following.

**Localization problem.** Given a sensor network with an undirected graph $\mathcal{G}$ containing a collection of $N$ ordinary nodes and $M$ anchor nodes, and the distance measurements of each node to its neighbors, the goal is to produce a set of position estimates $\mathbf{r}_i$ that are consistent with the actual positions of all ordinary nodes, that is, an assignment of points $\mathbf{r}_i$ such that $\mathbf{r}_i = \mathbf{r}^*_i$ for $i \in \mathcal{N}$.
It is worthwhile to mention that although the assignment might not be unique in some cases where specific mirror symmetry occurs or the network is incompletely observable, the localization problem in this article only targets at the case with sufficient observability and a unique embedding assignment.

**MSR algorithm**

The MSR algorithm is recalled here in detail in order to help make its main principle clear. Although the MSR algorithm has different aliases (e.g. anchor-free localization algorithm) in the research literature, its principle remains the same. In the MSR algorithm, the nodes are treated as distinct virtual masses, and the resulting distances among the masses are treated as a group of springs that connect neighbor nodes. Besides, the node position estimates are considered as the real-time positions of the masses, and the distance measurements are considered as the unstressed lengths of the springs. For every mass $i$ in such a virtual first-order mass–spring system, non-zero repulsive or attractive velocities that contributed by its neighbors will act on it immediately as long as the springs are deformed, that is, when their resulting distances are not equal to the unstressed lengths. Therefore, the resultant velocity of each mass will be obtained by combining all the component velocities that are contributed by its neighbors and drives each mass to move. The motion process will continue until nodes get a balance, which is regarded as the solution to the localization problem.

In a mathematical expression, the MSR algorithm,\textsuperscript{9,17–20} updates the position estimate for each node following

$$r_i^{[k]} = r_i^{[k-1]} + \sum_{j \in N_i \cup M} \left\{ a_{ij} \cdot \gamma \cdot n_{ij}^{[k-1]} \cdot \left( |r_i^{[k-1]} - r_j^{[k-1]}| - d_{ij}^{[k-1]} \right) \right\}$$  \hspace{1cm} (1)$$

where the superscript $[k]$ indicates the moment $k$; $|| \cdot ||$ is the vector norm operator; $\gamma$ is a proper positive scalar gain; $r_i$, $a_{ij}$, and $d_{ij}$ are defined the same as before; and $n_{ij}$ is the unit direction from $r_i$ to $r_j$, given by $n_{ij} = (r_j - r_i) / |r_j - r_i|$. 

**Performance metric for localization errors**

**Global energy ratio.** The global energy ratio (GER) is used to indicate the structural error of the localization, given by\textsuperscript{\textsuperscript{9}}

$$\text{GER} = \sqrt{\frac{2}{N(N-1)} \sum_{1 \leq i < j \leq N} \left( \frac{|r_i - r_j| - |r_i' - r_j'|}{|r_i' - r_j'|} \right)^2}$$  \hspace{1cm} (2)$$

Figure 2. An illustration of how to produce computational topologies upon the physical topology. As the imaginary switches are turned on or turned off, the information interaction among the nodes will be changed, yielding corresponding computational topologies.

The smaller the GER is, the better performance the localization would achieve. In the absence of measurement noises, the global minimum of the localization always corresponds to a GER value near 0.

**TCMSR algorithm**

In this part, we develop a useful interacting mechanism for each node at first, give a brief explanation on the role of the mechanism after that, and present the TCMSR algorithm at last.

**Interacting mechanism**

The mechanism is such established that for all nodes in a sensor network, interaction no longer follows the physical topology itself, but the computational topologies that are produced upon it.

Figure 2 illustrates the way of producing computational topologies upon the physical topology, where a series of controlled switches are imaginarily mounted on all the edges of the physical topology. Note that each undirected edge will be considered as two independent directed edges and be mounted by two independent switches. For example, suppose that nodes $i$ and $j$ have an undirected edge in the physical topology, then we regard the edge as two reversely directed edges $e_{ij}$ and $e_{ji}$ that start from one node and point to one another, and two independent switches termed as $s_{ij}$ and $s_{ji}$ will be respectively mounted on the directed edges. The connectivity of each directed edge $e_{ij}$ relies on the switch value $s_{ij}$. With $s_{ij}$ taking different values (0 and 1) for any $i$ and $j$, corresponding computational topologies can be produced.

Denote by a series of $\mathcal{G}$ the graphs of the computational topologies, by $\mathcal{A} = [\bar{a}_{ij}]$ the adjacency matrix
associated with $\mathcal{G}$, and by $\mathcal{L} = [I_{ij}]$ the Laplacian matrix. The new neighbor set consisting of all the neighboring nodes of $i$ in $\mathcal{G}$ is denoted by $\Omega_i$. It is clear that each $\mathcal{G}$ is a subgraph of $\mathcal{G}$. Moreover, $\hat{a}_{ij} = s_{ij} \cdot a_{ij}$, $\hat{l}_{ij} = s_{ij} \cdot l_{ij}$, and $\Omega_i \subseteq \mathcal{N}_i$.

In this article, $s_{ij}$ is a Boolean variable that is dynamically determined by

$$s_{ij} = \begin{cases} 1 & \text{if } \text{rand}() > \tau_i \\ 0 & \text{otherwise} \end{cases}$$

(3)

here, rand() is a random function that generates random float numbers within $(0,1)$. $0 < \tau_i < 1$ is a proper parameter. It can be noted that the greater $\tau_i$ is, the higher probability for $s_{ij}$ to take the value 0.

The interacting mechanism developed in this article can be then known as the rule that nodes interact following $\mathcal{G}$ instead of $\mathcal{G}$.

**Brief explanation**

Although the introduction of the mechanism seems trivial, its nontrivial contribution lies exactly in that the mechanism can effectively save the localization from being trapped in local minima. To explain this, we look into equation (1) and observe what has happened in local minima. For brevity, we simply refer to the term $n_{ij} \cdot \hat{(r}_i - r_j \hat{)} - d_{ij}$ in equation (1) as $v_{ij}$, which implies the component velocity acted on node $i$ that is contributed by node $j$.

First of all, we would like to explain the exact reason why traditional MSR may be trapped in local minima. Equation (1) expects that for any node $i$, the estimate should be driven by the component velocities until it arrives at the correct position, where $v_{ij} = 0$ holds for all $j$. However, equation (1) reveals that the equilibrium is not unique when all $v_{ij} = 0$. This situation, to the best of the authors’ knowledge, makes the occurrence of local minima. Actually, it is clear that local minima only when $\sum_{j \in \mathcal{N}_i} v_{ij} = 0$ while $\sum_{j \in \mathcal{N}_i} |v_{ij}| \neq 0$.

For helping better understand the reason explained above, we take Figure 6 as an instantiation of local-minima-trapped localizations. Without loss of generality, focus on a specific node $i = 25$, and list all its component velocities in Table 1. It is clear that not all the component velocities equal to 0 while the summation of the velocities equals to 0 approximately. As a result, the estimate $r_i$ cannot be further refined using equation (1), implying that the estimation has fallen into the local minimum.

Next, we explain why our mechanism can easily break up the trap. By selecting a proper $\tau_i < 1$, there must be a positive probability for some $s_{ij}$ to be zero. As a result, the event when $\Omega_i \neq \mathcal{N}_i$ must happen with a positive probability. Since $\Omega_i$ may be any subset of $\mathcal{N}_i$, there always exists a positive probability for some $\Omega_i$ to make the resultant velocity of node $i$ be unequal to zero, that is, $\sum_{j \in \Omega_i} v_{ij} \neq 0$ (despite that $\sum_{j \in \mathcal{N}_i} v_{ij} = 0$). Therefore, the local-minima equilibrium turns to be unstable immediately, and hence, the localization gets a chance to jump out of the trap. This explanation shows the validness of our mechanism briefly, and a more detailed proof will be provided in the following section.

**Formal proposal of the algorithm**

The TCMSR algorithm can be straightly proposed as

$$r_i^{[k]} = r_i^{[k-1]} + \sum_{j \in \mathcal{N}_i} \left\{ \hat{a}_{ij} \cdot \gamma_i^{[k]} \cdot n_{ij}^{[k-1]} \cdot \left( |r_i^{[k-1]} - r_j^{[k-1]}| - d_{ij}^{[k-1]} \right) \right\}$$

(4)

where $[k]$, $\hat{\cdot}$, $r_i$, $a_{ij}$, $d_{ij}$, and $n_{ij}$ are defined previously in equation (1), $\hat{a}_{ij} = s_{ij} \cdot a_{ij}$, $s_{ij}$ is determined by equation (3), and time varying $\gamma_i$ satisfies $0 < \gamma_i < (1/|\Omega_i|)$, with $|\Omega_i| = \sum_{j \in \Omega_i} \hat{a}_{ij}$ being the number of computational neighbors of node $i$ at each time slice.

Before going on, some lemmas are given as follows.

**Lemma 1.** Let $S_1, S_2, \ldots, S_k$ be a finite set of stochastic, indecomposable, and aperiodic\(^{28,29}\) (SIA) matrices. Then, there exists a column vector $y$ such that $\lim_{k \to \infty} S_1 S_2 \cdots S_k = 1 \cdot y^T$.

**Lemma 2.** If the union of the set of simple graphs $\{G_1, G_2, \ldots, G_m\} \subseteq \mathcal{G}$ has a spanning tree, then the matrix product $D_m \cdots D_2 D_1$ is SIA, where $D_i$ is a stochastic matrix corresponding to each simple graph $G_i$.

**Lemma 3.** Any local minimum will be unstable under algorithm (equation (4)).
Proof. The proof can be simply performed by reduction to the absurd. Suppose that there exists a stable local minimum at time slice $k$. This implies that $\sum_{i \in \Omega_k} v_{ij} = 0$ holds for all $i$ and $j$, while $\sum_{j \in \mathbb{N}} v_{ij} \neq 0$. However, there must be such a coming slice $k'$ that $\sum_{i \in \Omega_k} v_{ij} = 0$ no longer holds for all $i$ and $j$, with $\Omega_k^{k} \neq \Omega_k^{k'}$. This conflicts with the assumption that the local minimum is stable. Therefore, the proof can be completed by concluding from the absurd.

Lemma 4. The global minimum stays stable under algorithm (equation (4)).

Proof. For the global minimum, $v_{ij} = 0$ holds for all $i$ and $j$. Therefore, it is clear that for any time slice and $\Omega_i$, it always holds that $\sum_{j \in \Omega_i} v_{ij} = 0$, which implies that the global minimum stays stable. This completes the proof.

With these lemmas, the following theorem can be studied.

Theorem 1. For a localization problem with an initial guess as the starting point, the global minimum of the localization can be finally achieved with probability 1 using the TCMSR algorithm (equation (4)) for enough iterations.

Proof. First of all, we show that the algorithm can converge at least to some minima. At any time, each node $i$ has a current estimate $r_i$ on its position. Each node $i$ also periodically sends this position estimate to all its neighbors. Therefore, each node knows its own estimated position and the estimated positions of all its neighbors. Using these position estimates, each node $i$ calculates the estimated distance $|r_i - r_j|$ to each neighbor $j$. It also knows the measured distance $d_{ij}$ to each neighbor $j$.

The difference between the estimated distance and the measured distance immediately generates either an attractive or a repulsive velocity $v_{ij} = n_{ij}(|r_i - r_j| - d_{ij})$ that acts on node $i$ in the direction $n_{ij}$, with $n_{ij} = (r_j - r_i)/|r_j - r_i|$ being the unit vector in the direction from $r_i$ to $r_j$. Note that only when $s_{ij} > 0$ does each neighbor $j$ contribute a component velocity $v_{ij}$ to node $i$, and the resultant velocity $v_i$ acting on each node $i$ is the composition of all the contributory component velocities.

The structural energy $E_{ij}$ of nodes $i$ and $j$ due to the difference in the measured and estimated distances is the square of the difference, that is, $E_{ij} = (|r_i - r_j| - d_{ij})^2$, and the total energy of node $i$ equals $E_i = \sum_j E_{ij}$. The energy $E_i$ of each node $i$ reduces when it moves by a small amount $\Delta r_i = \alpha_i \gamma_i$ in the direction of the velocity $v_i$. Likewise, the total energy of the system $E = \sum_i E_i$ reduces as each node moves. This implies that the algorithm can converge with a small $\gamma_i$. Note that the small parameter $\gamma_i$ scales the exact amount $\Delta r_i$ by which each node $i$ moves. To determine the value range of $\gamma_i$, consider the local convergence near some converged minimum.

Denoting $h_{ij} = -n_{ij} \cdot d_{ij}$ (regardless of the zero-denominator singularity) and noting that $r_i - r_j = -n_{ij}|r_j - r_i|$, one can rewrite equation (4) as

$$r_{ij}^{k} = r_{ij}^{k-1} - \sum_{j \in \mathbb{N}} \alpha_{ij} \cdot \gamma_{ij} (r_{ij}^{k-1} - r_{ij}^{k-1} - h_{ij}^{k-1})$$

(5)

At any time, one can always treat $h_{ij}$ as $h_i = h_i - h_j$, with $h_i$ being a proper vector for any $i \in \mathbb{N}$. In particular, let $h_i = p_i$ for $i \in \mathbb{M}$ since anchor nodes are aware of their actual positions. By introducing the error state $\tilde{r}_i = r_i - h_i$ for any $i$, one can further rewrite equation (5) in the error-form equation as

$$\tilde{r}_i^{k} = \tilde{r}_i^{k-1} - \sum_{j \in \mathbb{N}} \alpha_{ij} \cdot \gamma_{ij} \cdot (\tilde{r}_i^{k-1} - \tilde{r}_j^{k-1}) + h_i^{k-1} - h_i^{k}$$

(6)

With the observation that $h_i$ changes much more slowly than $r_i$, as the algorithm converges (by observing $h_{ij} = -n_{ij} \cdot d_{ij}$), one have that $\Delta h_i = o(r_i - r_j)$ as $k$ increases. Therefore, around the optima equation (6) can be simplified with a good approximation as

$$\tilde{r}_i^{k} = \tilde{r}_i^{k-1} - \sum_{j \in \mathbb{N}} \alpha_{ij} \cdot \gamma_{ij} \cdot (\tilde{r}_i^{k-1} - \tilde{r}_j^{k-1})$$

(7)

Since equation (7) is component-wise in form, one can just consider it in a one-dimensional case, and the consideration of other dimensions is the same. Therefore, equation (7) can be rewritten in the compact matrix form, yielding

$$\tilde{r}^{k} = D^{k} \tilde{r}^{k-1}$$

(8)

where $\tilde{r} = [\tilde{r}_1, \tilde{r}_2, \ldots, \tilde{r}_N + M]^T$, $D^{k} = (I - Y^{k} \cdot L)^{k}$, $Y^{k} = \text{diag}(\gamma_1^{k}, \ldots, \gamma_N^{k} + M)$, $L$ is the Laplacian matrix associated with the graph $\mathcal{G}$, with $\mathcal{G}$ being the graph of the computational topology.

Due to the fact that the physical graph $\mathcal{G}$ contains a spanning tree (since the given localization problem is sufficiently observable), the probability for a finite union $\{\tilde{G}^{k_1}, \tilde{G}^{k_2}, \ldots, \tilde{G}^{k_M}\}$ to have a directed spanning tree equals to 1. Therefore, a period of time slices $[0, m_1]$ can be divided into a sequence of segments $[0, m_1], [m_1 + 1, m_2], \ldots, [m_{l-1} + 1, m_l]$, so that the union of the
Lemma 2 that any computational topology for all and i.

Denote by \( S^{(l)} \), the disk theorem indicates that \( D^{(k)} \) is stochastic. As a result, one obtains the fact from Lemma 2 that \( S^{(l)} \) is SIA for any segment \( l \). Following Lemma 1, the error states converge, that is

\[
\lim_{l \to \infty} \mathbf{E} = \lim_{l \to \infty} \sum_{i=0}^{d} \mathbf{S}^{A} \mathbf{S}^{(l)} \mathbf{F}^{(i)} = \mathbf{1} \cdot e
\]

with \( e \) being a constant. It means that the error states converge to achieve a consensus, that is, \( \mathbf{E} \to 0 \) for any \( i \) and \( j \). By noting that \( \mathbf{E} \to 0 \) holds for \( i \in \mathcal{M} \), one finally obtains \( \mathbf{E} \to 0 \) and equivalently \( r_i \to h_i \) for any \( i \). As above all, it is shown that the algorithm can converge to achieve a consensus, that is, each \( r_i \) in equa-

Second, Lemma 3 points out the fact that any local minimum will be unstable under the proposed algo-

rithm. Since the computational topology is randomly produced upon the physical topology, the motion trajec-

tory escaping from the current minimum would vary with different iterations and thus will reset varying

starting points for the next localization epoch.

Third, since the starting points of the localization vary after local minima are escaped from, the probabil-

ity for a single localization epoch to converge to the glo-

bal minimum, termed as \( P \), is larger than zero, that is, \( P > 0 \). Therefore, after enough time of epochs, the final probability for the localization to converge to the global minimum, termed as \( P_{\text{global}} \), can be obtained as

\[
P_{\text{global}} = \lim_{q \to \infty} [1 - (1 - P)^q] = 1
\]

This indicates that the global minimum can be event-

ually achieved using the algorithm.

Finally, Lemma 4 shows that the global minimum will be stable whenever it is achieved. Above all, the proof has been completed.

Additional discussion

First, we would like to discuss the parameter \( \tau_i \) in equa-

tion (3). Denote by \( \Omega_i \) the neighbor set of node \( i \) in a computational topology for all \( i \) and by \( |\cdot| \) the card of the set. Therefore, the number of neighbors of node \( i \) in a specific computational topology \( \mathcal{G} \) can be termed by \( |\Omega_i| \). Denote by \( \bar{\Omega}_i \) the mathematical expectation of \( |\Omega_i| \) across all \( \mathcal{G} \), then equation (3) indicates that

\[
|\Omega_i| = (1 - \tau_i) \cdot |N_i|.
\]

It is clear that the problem of choosing an optimal \( \tau_i \) can be equivalent to the dual problem of choosing an optimal \( \bar{\Omega}_i \).

To the best knowledge of the authors, there is hardly no mathematical method to exactly solve both the pri-

mal and the dual problems. In other words, the prob-

lem of choosing an optimal parameter remains unsolved and needs further research.

In this article, we temporarily provide an empirical way to help determine a workable value for the para-

meter. On one hand, by recalling the reason of the trap of local minima, we know that the trap takes place because the neighboring components counteract each other. It is then clear that for each node, the degree of counteraction will become heavier when the number of neighbors gets large (in analogy with molecular thermal movement), and hence, a smaller \( \bar{\Omega}_i \) is expected to help localization escape from local minima. While on the other hand, a too-small \( \bar{\Omega}_i \) will reduce the connectivity of computational networks and is harmful to conver-

gence. Concluding from both hands, one can obtain a guideline as: a relatively larger \( \bar{\Omega}_i \) is beneficial to convergence but is harmful to searching for global opti-

mization (e.g. when \( \bar{\Omega}_i = |N_i| \), TCMSR will com-
pletely degrade into MSR), while a relatively smaller \( \bar{\Omega}_i \) is beneficial to searching for global optimization but is harmful to convergence. As a tradeoff, we empiri-

cally consider \( 1 \leq \bar{\Omega}_i \leq 2 \) to be of benefit. For the simulations of this research, in particular, we capri-

ciously let \( \bar{\Omega}_i = 2 \) for every node \( i \), and now that

\[
\tau_i = 1 - \frac{2}{|N_i|}
\]

Second, it is worthy of mentioning that the time complexity of TCMSR is not higher than that of MSR and has no relation with the size of sensor networks. Compared to MSR, TCMSR introduces no extra action in the communication process and uses less neighboring information to take part in the updating process. The only introduced action is the lightweight generation of \( s_{ij} \) for each computational neighbor. It means that TCMSR can be applicable in all scenarios where MSR can be used. However, it should also be mentioned that TCMSR guarantees global optimization only under the assumption of enough iterations, so that in some real scenarios where only finite iterations are allowed, TCMSR may not always achieve global optimization.

Simulations

In this section, we verify the proposed TCMSR algo-

rithm by numerical simulations and compare it with the traditional MSR algorithm. The only parameter \( \gamma \) in the MSR is set to be \( 1/2|N_i| \) for each node \( i \) as reported, and the main parameters in the proposed TCMSR algorithm are set as described in Theorem 1,
that is, \( t_i = 1 - (2/|N_i|) \), \( \gamma_i = 1/(|\Omega_i| + \theta) \), \( |\Omega_i| = \sum_{j \in \Omega_i} \), \((s_{ij}, a_{ij})\) is self aware by each node and updates with iterations, and \( \theta = 0.01 \) is a small positive tolerance to guarantee \( \gamma_i < 1/|\Omega_i| \). In total, three experiments are performed.

**Specific case: with a fold-free (good) starting point**

In the first experiment, we compare the MSR and the TCMSR with a fold-free starting point in an instantiation of sensor networks. The sensor network is deployed in a 400 units by 200 units (unit of length, for example, centimeter, meter, and kilometer) two-dimensional square space, where \( M = 4 \) anchor nodes are placed at the four corners and \( N = 150 \) ordinary nodes are placed at distinct random locations. The physical topology is generated with a mean connectivity \( J_{mean} = 16.22 \) by letting the maximal communication radius be 60 units. In fact, measuring noises would be related with specific measuring principles and many factors. However, considering that noises are not a major issue for the particular problem discussed in this article, we take no account of any physical models to describe the measuring noises and treat the noise as a Gaussian distribution simply. In particular, distance measurements among neighboring nodes are generated with Gaussian noises added, whose \( 3\sigma \) boundaries are all 0.03 unit. Figure 3 shows the actual position assignment and the interaction for the instantiation.

Both MSR and TCMSR are given a same fold-free starting point, which is actually in a close neighborhood of the correct assignment. The localizations are then performed using the two algorithms, respectively, and their results are plotted simultaneously in Figure 4. It can be seen that both the two algorithms can successfully make the localization converge to the global minimum (i.e. the correct assignment). Despite that the performances of the two algorithms seem similar, the slight performance difference between the two algorithms lies exactly in the feature that TCMSR would jitter stronger than MSR does when they are converging. This is because TCMSR uses computational topologies to interact among nodes instead of the fixed physical topology, and the switching of computational topologies cause the jitter. However, the jitter reflects that TCMSR behaves in a more aggressive way to converge. It can be also seen that the jitter of TCMSR weakens and disappears after the algorithm has converged. This shows the fact that the global minimum would not suffer from the proposed TCMSR algorithm.

**Specific case: with a random (bad) starting point**

In the second experiment, we compare MSR and TCMSR with a purely random starting point. The same instantiation of sensor networks is used, as described in the first experiment. Different from the first experiment, a random guess on position assignment, which has a significant estimation error, is set as the starting points of both the two algorithms. Localization results are shown in Figure 5. It can be seen that MSR finally makes the localization converge to and gets trapped in a local minimum with a significant bias, implying that the traditional MSR fails in the localization. For a more visualized viewing, Figure 6 illustrates the localization errors in a two-dimensional view. The dot dash lines clearly indicate that MSR converged to a local minimum. However, the localization estimation cannot be refined anymore. This is because the resultant velocity acting on each node has become extremely small,
although not all the component velocities are small. Table 1 shows a more detail of component velocities of an arbitrary node $i = 25$.

On the contrary, the proposed TCMSR can help the localization converge to the global optimum, since the localization error converges to zero successfully. This success is guaranteed by the effective mechanism, under which local minima can never stay stable. After enough time of iterations, the global minimum can be eventually achieved by using TCMSR.

Monte Carlo simulation

In the third experiment, we perform the Monte Carlo simulation for MSR and TCMSR, respectively. In total, 300 different sensor networks are generated by varying the number of ordinary nodes ($50 \leq N \leq 300$) and the mean connectivity degree ($8 \leq J_{\text{mean}} \leq 40$). For each generated sensor network, three different starting points are randomly guessed for both the algorithms. Therefore, for both the algorithms, 900 times of localizations are performed.

Results of the simulation present that the traditional MSR algorithm fails in about 60% of the localizations, showing a high sensitivity to network conditions and starting points. On the contrary, the proposed TCMSR algorithm can always achieve the global minimum for the localization in all the simulations, even starting from a purely random starting point (on the same scale of the correct assignment). The success rate using TCMSR is an exact 100%.

As a summary, the simulation shows that the proposed TCMSR algorithm can work not only with the fold-free starting point but also with the purely random starting point. Compared to the traditional MSR algorithm, TCMSR behaves better in achieving the global minimum, with a more effective and robust performance. Considering the fact that there is no simple way to ensure a good enough starting point, the proposed TCMSR algorithm shows a great advantage for practical localization problems.

Conclusion

This article deals with the distributed localization problem in wireless sensor networks. Traditional distributed algorithms for this problem are likely to suffer from the local-minima failures. Focusing on this issue, a fully distributed algorithm is proposed under a novel mechanism, where each individual node is allowed to computationally interact with a random subset of its neighbors for helping localizations escape from local minima. Theoretical analyses reveal that any local minimum of the localization will be unstable under the proposed algorithm, and that the global optimum would finally be achieved with probability 1 after enough time of iterations. Numerical simulations given at last have demonstrated the effectiveness of the algorithm.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Natural Science Foundation of China under Grants 61701498, 61605099, and 61703403, the Young Elite Scientists Sponsorship Program by CAST (2016QNRC001), the Youth Talent Program of Beijing High-level Innovation and
Entrepreneurship under Grant G04070017, the National Key R&D Program of China under Grant 2018YFA0703800, and the Youth Talent Program of Beijing High-level Teachers under Grant 201704062. Co-Corresponding author should be addressed to all the authors.

ORCID ID
Xiaochu Wang https://orcid.org/0000-0002-8929-9920

References

1. Bal M, Liu M, Shen W, et al. Localization in cooperative wireless sensor networks: a review. In: 13th international conference on computer supported cooperative work in design CSCWD, Santiago, Chile, 22–24 April 2009, pp.438–443. New York: IEEE.
2. Yassin A, Nasser Y, Awad M, et al. Recent advances in indoor localization: a survey on theoretical approaches and applications. IEEE Commun Surv Tut 2016; 19(2): 1327–1346.
3. Pal A. Localization algorithms in wireless sensor networks: current approaches and future challenges. Netw Protoc Algor 2010;2(1):45–73.
4. Panwar A and Kumar SA. Localization schemes in wireless sensor networks. In: 2012 second international conference on advanced computing & communication technologies (ACCT), Rohtak, India, 7–8 January 2012, pp.443–449. New York: IEEE.
5. Hossain AM and Soh W-S. A survey of calibration-free indoor positioning systems. Comput Commun 2015; 66:1–13.
6. Paul A and Sato T. Localization in wireless sensor networks: a survey on algorithms, measurement techniques, applications and challenges. J Sens Actuator Netw 2017; 6(4): 24.
7. Van Nguyen T, Jeong Y, Shin H, et al. Least square cooperative localization. IEEE Trans Veh Technol 2015; 64(4): 1318–1330.
8. Han G, Xu H, Duong TQ, et al. Localization algorithms of wireless sensor networks: a survey. Telecommun Syst 2013; 52(4): 2419–2436.
9. Priyantha NB, Balakrishnan H, Demaine E, et al. Anchor-free distributed localization in sensor networks. In: Proceedings of the 1st international conference on embedded networked sensor systems, Los Angeles, CA, 5–7 November 2003, pp.340–341. New York: ACM.
10. Khan U, Kar S, Moura JM, et al. Distributed sensor localization in random environments using minimal number of anchor nodes. IEEE Trans Signal Process 2009; 57(5): 2000–2016.
11. Zhu S and Ding Z. Distributed cooperative localization of wireless sensor networks with convex hull constraint. IEEE Trans Wirel Commun 2011; 10(7): 2150–2161.
12. Wang X, Sun T and Fan C. Neighborhood Kalman estimation for distributed localization in wireless sensor networks. Math Probl Eng 2016; 2016: 1–8.
13. Chu H and Wu C-D. A Kalman framework based mobile node localization in rough environment using wireless sensor network. Int J Distrib Sensor Netw 2015; 11(5): 841462.
14. Cattivelli FS and Sayed AH. Distributed nonlinear Kalman filtering with applications to wireless localization. In: 2010 IEEE international conference on acoustic, speech and signal processing (ICASSP), Dallas, TX, 14–19 March 2010, pp.3522–3525. New York: IEEE.
15. Wang X, Zhao K and You Z. Coordinated motion control of distributed spacecraft with relative state estimation. J Aerosp Eng 2016; 29(3): 04015068.
16. Wymeersch H, Lien J and Win MZ. Cooperative localization in wireless networks. Proc IEEE 2009; 97(2): 427–450.
17. Xiao Q, Xiao B, Bu K, et al. Iterative localization of wireless sensor networks: an accurate and robust approach. IEEE/ACM Trans Netw 2014; 22(2): 608–621.
18. Howard A, Mataric MJ and Sukhatme GM. Relaxation on a mesh: a formalism for generalized localization. In: IEEE international conference on intelligent robots and systems, Maui, HI, 29 October–3 November 2001, pp.1055–1060. New York: IEEE.
19. Gotsman C and Koren Y. Distributed graph layout for sensor networks. In: International symposium on graph drawing, Perugia, 21–24 September 2004, pp.273–284. Berlin: Springer.
20. Wu H, Wang C and Tzeng N-F. Novel self-configurable positioning technique for multihop wireless networks. IEEE/ACM Trans Netw 2005; 13(3): 609–621.
21. Abouzar P, Michelson DG and Hamdi M. RSSI-based distributed self-localization for wireless sensor networks used in precision agriculture. IEEE Trans Wirel Commun 2016; 15(10): 6638–6650.
22. Naraghi-Pour M and Rojas GC. A novel algorithm for distributed localization in wireless sensor networks. ACM Trans Sensor Netw 2014; 11(1): 1.
23. Wang X, Sun C, Sun T, et al. Distributed lost-in-space localization in sensor networks using range measurements. In: 2018 IEEE international conference on systems, man, and cybernetics (SMC), Miyazaki, Japan, 7–10 October 2018, pp.2912–2916. New York: IEEE.
24. Priyantha NB, Balakrishnan H, Demaine ED, et al. Mobile-assisted localization in wireless sensor networks. In: Proceedings IEEE 24th annual joint conference of the IEEE computer and communications societies, Miami, FL, 13–17 March 2005, pp.172–183. New York: IEEE.
25. Han G, Jiang J, Zhang C, et al. A survey on mobile anchor node assisted localization in wireless sensor networks. IEEE Commun Surv Tutor 2016; 18(3): 2220–2243.
26. Halder S and Ghosal A. A survey on mobility-assisted localization techniques in wireless sensor networks. J Netw Comput Appl 2016; 60: 82–94.
27. Eren T. Cooperative localization in wireless ad hoc and sensor networks using hybrid distance and bearing (angle of arrival) measurements. EURASIP J Wirel Commun Netw 2011; 2011(1): 72.
28. Ren W and Beard RW. Consensus seeking in multiagent systems under dynamically changing interaction topologies. IEEE Trans Autom Control 2005; 50(5): 655–661.
29. Wollowitz J. Products of indecomposable, aperiodic, stochastic matrices. Proc Am Math Soc 1963; 14(5): 733–737.
30. Horn RA and Johnson CR. Matrix analysis. Cambridge: Cambridge University Press, 2012.