Locally Estimating Log-normal Shadowing Parameters with Applications in Graph Planarization

Steffen Böhmer
University of Koblenz, Germany
steffenboehmer@uni-koblenz.de

Sven-Niklas Wollny
University of Koblenz, Germany
snwollny@uni-koblenz.de

Hannes Frey
University of Koblenz, Germany
frey@uni-koblenz.de

***Abstract*—We consider wireless networks with homogeneously Poisson distributed node positions and node connections based on the log-normal shadowing model. We describe a local method for estimating the log-normal shadowing parameters and the intensity of the underlying Poisson point process which are both global network parameters. We then apply the locally found parameters in a local method to planarize wireless network graphs which so far was depending on knowledge of these global parameters. Thereby, we present with this paper a fully local planarization strategy for wireless communication networks well beyond simplified wireless network models. Here, planarization means constructing a subgraph of a graph with nodes on the plane, such that the subgraph has no intersecting edges. At the same time, edges are to be removed in such a way that the subgraph remains connected. Local estimation of the log-normal shadowing parameters as well as local planarization based on these estimated parameters are then investigated by simulation. The simulation results show that local estimation of the parameters in combination with local planarization almost achieves the performance of the so far known method based on global knowledge.**

I. Introduction

Many cyber-physical systems such as distributed sensor networks, distributed swarms of robots, collaborating fleets of autonomous vehicles or ad-hoc networked user devices consist of a large number of nodes forming a wireless network that is supported by the network interfaces of the nodes themselves.

Applications in such networked systems like some localization, distributed control, topology control or message forwarding methods can be based on models of the underlying propagation model. When utilizing propagation models, it is sometimes required to know the parameters of the model. We focus in this work on the log-normal shadowing model. We are interested in estimating the model parameters which are the path loss coefficient and the shadowing variation. In addition, given this estimate we also infer a further parameter, the general node density of the network.

We then use the log-normal shadowing parameter estimates as input to the improved EAR-algorithm from [1]. That algorithm is intended to locally planarize a graph, i.e. to find a connected subgraph without intersecting links. Such planar drawings have their applications in different local network algorithms as for example discussed in [2]. By local we term algorithms, that each network participant can perform solely based on information about its neighborhood instead of global information about the whole network.

When applied under log-normal shadowing, the improved EAR-algorithm from [1] yields an almost connected subgraph with almost no intersecting links. However, it requires global information in terms of the exact log-normal shadowing parameters. Here, we show that the algorithm can be formulated fully local using our local parameter estimations. With that local variant connectivity is slightly improved, however, at the cost of a few more remaining intersecting links.

To summarize our contribution, we propose a strategy to estimate key parameters of the log-normal shadowing model that opposed to existing work requires solely information about edge lengths in a nodes neighborhood and yields all key parameters along with the networks' density. Moreover, for the first time a fully local planarization strategy for log-normal shadowing generated networks yielding almost connected and intersection free results is proposed.

The remainder of this paper is structured as follows. In the next section we introduce log-normal shadowing, Poisson point processes, and the graph planarization method studied in this work. We then present in Section III how log-normal shadowing parameters can be estimated just based on local network observations. In Section IV we then discuss empirical simulation results on precision obtained by our proposed method. In a separate Section V we then investigate our estimation method in connection with the EAR-algorithm. We compare by simulation the performance assuming perfect knowledge about the log-normal shadowing parameters to the performance when using knowledge obtained by our local method. Finally, in Section VI we conclude our findings discussing how far the described local estimates are usable for the local planarization strategy studied here and how far the local parameter estimation can be extended beyond the model assumptions studied in this work.

II. Theoretical Foundations, Problem Formulation and Related work

A. Mathematical Foundations

In this work we consider random graphs to model wireless networks. We utilize a homogeneous Poisson point process (PPP) with node density λ to sample a set of network nodes.

To model communication links we consider outage probabilities under log-normal shadowing (LNS). Given path loss coefficient α, communication energy loss threshold β in dBm (i.e. the maximum allowed signal strength loss at the receiver)

and signal strength standard deviation σ, the probability that two devices with distance d can communicate with each other, is given by [3]:

$$p_{LNS}(d) := 1/2 - 1/2 \operatorname{erf}\left(10/\sqrt{2}\ \omega^{-1} \log\left(d/r\right)\right) \quad (1)$$

using the auxiliary parameters $r = 10^{\beta/10\alpha\mathrm{dB}}$ denoting the theoretical communication distance (without signal strength deviation) and the ratio between signal deviation and path loss $\omega = \sigma/\alpha$.

The limiting case $\omega = 0$ is equivalent to the unit disk graph model (UDG) with radius r, also known as Gilbert disk model, in which an edge between two nodes exists iff their distance is less or equal to r.

B. Joint LNS-Parameter and Density Estimation Problem

Given a (sub-)graph with nodes obtained from a homogeneous PPP and edges sampled corresponding to a LNS connection probability function, the task is to estimate the underlying global parameters. We assume that β is a system constant, and hence known. So the task reduces to estimating σ, α and λ. Since α is in a one to one correspondence with r (for fixed β) and σ is in a one to one correspondence with ω (for known α) the task is equivalent to estimating the triple (r, λ, ω).

In related work, given the global information of all distances and signal strength measurements, the shadowing parameters can be estimated in a standard way by linear regression [4]. Recently machine learning techniques are used to estimate the path loss parameters for environments, e.g. by interpreting images [5]. Many pseudo local estimations consider prior measurements taken at known positions or reference distances in addition to locally obtained information [6]. All of the methods mentioned have in common that at least partially a global view of the graph is required and thus are not suitable for use cases, where such global information is not available. Local estimation of just the path loss coefficient was addressed in numerous publications. In [7] the path loss coefficient assuming Nakagami fading was estimated pseudo locally assuming knowledge of either the node density or assuming to change the receiver's sensitivity. In [8] the latter problem was addressed and a fully local estimation of the path loss coefficient via linear regression was presented. In [9] the path loss coefficient and signal strength deviation are estimated by signal strength measurements taken over cliques of four nodes. Although technically locally applicable, the proposed estimation only yields stable results with global information. The node density of the network is not addressed in neither of the mentioned publications. However, for applications like the following all parameters of the graph model are needed. A local approach to determine them all is presented in this work.

C. Application

In this section we will test our estimation strategy in the application field of topology control. In [10] the EAR-algorithm was developed as an advancement of the relative neighborhood graph to remove intersections from random connection graphs. In [1] it was combined with a technique called cutoff to planarize LNS-graphs without harming the connectedness significantly.

The planarization strategy works as follows: From each triangle or quadrilateral in the original graph the longest edge is removed from the graph. We term this strategy EAR-algorithm or just EAR for short in the following.

The EAR-algorithm can be performed distributed with 2-hop information, i.e., every node just needs to know the part of the graph, that can be reached with at most two hops. It was shown [10], that the heuristic does not change the connected components of a graph and each UDG graph is planarized.

However, for the LNS model that does not hold in general. It was shown [10], that introducing a cutoff distance (i.e., deleting edges exceeding a certain length in graphs resp. turning off the links in real world networks) significantly increases the fraction of intersections being resolved. Since by using a cutoff the connectedness of the graph may change.

In large scale networks, i.e. spatial widely spread networks with many participants, a single node can be isolated from all others However, a large (connected) fraction of nodes in the same connected component is achievable. So we choose the number of nodes in the largest connected component to judge the connectedness of a large scale network. Such a component is called giant component (GC). Hence, the cutoff distance has to be chosen so that a sufficiently large number of nodes remain connected in the network graph.

In [1] a cutoff R based on a global parameter was proposed s.t. after application of this cutoff and subsequently the EAR-algorithm the graph is almost connected and contains almost no intersections:

$$R := (1 + 1/\omega) \inf\{x | p(x)x^2 = 1.437/\lambda\} \quad (2)$$

D. Notational conventions

By E, SD and M we term the stochastical expected value, standard deviation and median. By $\hat{E}$, $\hat{SD}$ and $\hat{M}$ we term the statistical equivalents. By $\tilde{\omega}, \tilde{r}, \tilde{\lambda}$ we term estimated values of real parameters ω, r, λ.

III. Estimation Strategy

For the moment we fix an ω and term $p_r^{(\omega)}(d)$ the LNS-probability function with theoretical communication distance r. Trivially we get from (1): $p_r^{(\omega)}(d) = p_1^{(\omega)}(d/r)$.

In [2] the probability density function of the length of a random edge $d_r^{(\omega)}$ in a graph obtained from a LNS-model over a homogeneous PPP was derived:

$$f_{d_r^{(\omega)}}(d) = \frac{d p_r^{(\omega)}(d)}{\int_0^\infty x p_r^{(\omega)}(x) \mathrm{d}x}$$

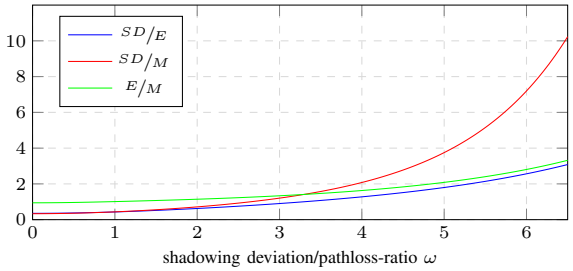


Fig. 1: Pairwise ratios of SD, E and M for $\omega \in [0, 6.5]$

So for the expected length of an edge $d_r^{(\omega)}$ we get

$$E[d_r^{(\omega)}] = \int_0^\infty d \frac{dp_r^{(\omega)}(d)}{\int_0^\infty x p_r^{(\omega)}(x)\mathrm{d}x}\mathrm{d}d = \frac{\int_0^\infty d^2 p_r^{(\omega)}(d)\mathrm{d}d}{\int_0^\infty x p_r^{(\omega)}(x)\mathrm{d}x}$$
$$= \frac{\int_0^\infty d^2 p_1^{(\omega)}(d/r)\mathrm{d}d}{\int_0^\infty x p_1^{(\omega)}(x/r)\mathrm{d}x} = r\frac{\int_0^\infty t^2 p_1^{(\omega)}(t)\mathrm{d}t}{\int_0^\infty t p_1^{(\omega)}(t)\mathrm{d}t} = rE[d_1^{(\omega)}]$$

substituting $x = rt$ resp. $d = rt$.

Similarly one gets for the standard deviation $SD[d_r^{(\omega)}] = rSD[d_1^{(\omega)}]$ and for the median $M[d_r^{(\omega)}] = rM[d_1^{(\omega)}]$. So the three ratios $SD[d_r^{(\omega)}]/E[d_r^{(\omega)}] = SD[d_1^{(\omega)}]/E[d_1^{(\omega)}]$, $SD[d_r^{(\omega)}]/M[d_r^{(\omega)}] = SD[d_1^{(\omega)}]/M[d_1^{(\omega)}]$ and $E[d_r^{(\omega)}]/M[d_r^{(\omega)}] = E[d_1^{(\omega)}]/M[d_1^{(\omega)}]$ just depend on ω and downgrade r to a scaling factor. Via numerical evaluation we suppose, that these ratios are injective for the application relevant ω-values up to 6.5 (see Figure 1).

So via evaluating the ratios of the experimentally determined statistical data $\hat{E}$, $\hat{SD}$ and $\hat{M}$, we can numerically approach up to three different $\tilde{\omega}_\bullet$ (If there is just one edge, $\hat{SD}$ cannot be evaluated. If there is no edge, no statistic can be defined). If a ratio is calculated to be out of the respective scope, we round too low ratios to the boundary value $\tilde{\omega}_\bullet = 0.0$ and too high ratios to $\tilde{\omega}_\bullet = 6.5$. We term the three different estimations $\tilde{\omega}_{SD/E}$, $\tilde{\omega}_{SD/M}$ and $\tilde{\omega}_{E/M}$.

The exact choice of $\tilde{\omega}$ from the three options will be discussed and determined in Section IV-B1

Once we fixed an $\tilde{\omega}$, we can calculate: $\tilde{r}_{SD} := \hat{SD}/SD[d_1^{(\tilde{\omega})}]$, $\tilde{r}_E := \hat{E}/E[d_1^{(\tilde{\omega})}]$ and $\tilde{r}_M := \hat{M}/M[d_1^{(\tilde{\omega})}]$

Again, we fix one of them as $\tilde{r}$ with the exact choice postponed to Section IV-B1.

In a PPP with density λ, the expected number of neighbors of a random node with connection probability function p can be calculated as [3]:

$$N(p) := 2\pi\lambda \int_0^\infty dp(d)\mathrm{d}d$$

So once we fixed $\tilde{\omega}$ and $\tilde{r}$ we chose $\tilde{\lambda} := \hat{N}/2\pi \int_0^\infty dp_{\tilde{r}}^{(\tilde{\omega})}(d)\mathrm{d}d$ with $\hat{N}$ being the statistical average number of neighbors of a node.

IV. Simulation of the Estimation Strategy

In this section we present an extensive simulation study to evaluate how precise our estimation strategy is.

At first, we just sample a large set of edge lengths of a LNS-graph to investigate how our estimator performs and which choice is the best among the three possible ω- and three possible r- estimations.

Afterwards we evaluate how well the estimation works if we localize the technique. Each node makes its own parameter estimation based on the local statistics taken from the neighborhood of the node. Since we already need 2-hop information for the EAR-algorithm, we perform the estimation with information from the 2-hop neighborhood of the nodes. We sample appropriate 2-hop neighborhoods of a single node to examine the local strategy in absence of boundary effects.

Finally, we run the estimation strategy on a proper sampled graph. We apply the strategy with both, global and local statistics. We use the global simulation to consider how good the estimation can get if one increases the size of the considered "local" subgraph and to detect e.g. boundary effects. The local results will be essential in the next section, where the fully local version of the EAR-algorithm needs local parameter estimations in a LNS-graph.

For the remainder of the paper we fix the simulation parameter settings $\lambda = 7.5 \times 10^{-5}$, $r = 200$ and $\omega \in \{0.0, 0.5, ..., 4.0\}$, following those from [1], which we compare the results of this paper against.

A. Simulation methods

1) Generated Distance Set: In our first simulation study we sample 10 sets of 100,000 edge lengths according to $f_{d_r^{(\omega)}}$.

This method does not include any local dependencies occurring in a graph. However, the distribution of edge lengths and hence the relevant statistics $\hat{SD}$, $\hat{E}$, $\hat{M}$ converge to the ones of a large or even infinite graph.

Note, that this method does not address any simulation area or node density.

2) 2-hop Neighborhood: Since the goal is to create a localized, parameter estimation scheme, next we test our estimation strategy in the 2-hop neighborhood of a node.

To generate the 2-hop neighborhood of a node without boundary effects, we run a randomized version of breadth first search (BFS). Instead of exploring the 2-hop neighborhood in a previously generated large enough graph, we generate the required nodes during the BFS-algorithm on demand.

This procedure was run for each ω-value 500 times.

3) Proper graph: Next we sample finite graphs to test our estimation strategy on large scale realistic networks. We fix an area of 12000×12000 (i.e., $60r \times 60r$). This results in 10800 nodes on average. A PPP is sampled and the nodes are connected via the LNS-model. For each ω-value 12 graphs are sampled.

Both estimation strategies are applied. The global estimation is based on the global statistics of the graph. In the local estimation each node runs it own estimation based on the statistics of its 2-hop neighborhood.

Note that, when applying the local estimation of λ (as well as in section IV-A2), $\hat{N}$ only takes the degree of the 1-hop neighborhood into account, since not all neighbors of a 2-hop

	$\tilde{\omega}$				$\tilde{r}$			
ω	$\tilde{\omega}_{SD/E}$	$\tilde{\omega}_{SD/M}$	$\tilde{\omega}_{E/M}$	Max	$\tilde{r}_E$	$\tilde{r}_M$	$\tilde{r}_{SD}$	Median
0.0	0.05	0.04	0.04	0.08	199.84	199.97	199.38	199.85
0.5	0.50	0.50	0.50	0.51	199.83	199.95	199.46	199.85
1.0	1.00	1.00	1.00	1.00	199.84	199.89	199.75	199.85
1.5	1.49	1.50	1.50	1.50	200.22	200.28	200.22	200.26
2.0	1.99	1.99	1.99	2.00	200.35	200.42	200.03	200.38
2.5	2.49	2.49	2.50	2.50	200.50	200.55	199.91	200.48
3.0	2.98	2.98	3.00	3.00	200.52	200.58	199.44	200.51
3.5	3.47	3.47	3.49	3.49	201.94	201.98	200.45	201.91
4.0	3.94	3.96	3.99	3.99	201.47	201.53	198.12	201.44

TABLE I: Estimated parameters for generated distances.

neighbor are explored in the 2-hop neighborhood and $\hat{N}$ would be skewed.

B. Results

1) Generated Distance Set: In this section we discuss the results from the estimation strategy applied on the edge set and based on that we determine the exact choice of $\tilde{\omega}$ and $\tilde{r}$.

Table I shows the results of this simulation, which will be discussed while fixing the estimation strategy in the follwing.

At first we have to fix an $\tilde{\omega}$ out of the three possible choices $\tilde{\omega}_{SD/E}$, $\tilde{\omega}_{SD/M}$ and $\tilde{\omega}_{E/M}$.

We can observe, that due to the huge sample size all estimations perform quite well. However, there are some larger discrepancies we can discuss.

On the one hand any $\tilde{\omega}_\bullet$ is a quite bad estimation for the UDG-case (at least 0.04 instead of 0). Statistical ratios of $\hat{SD}$, $\hat{E}$ and $\hat{M}$ below the corresponding y-intercept in Figure 1, that would suggest negative ω-values, are rounded to $\tilde{\omega}_\bullet = 0$. So they do not cancel out overestimated ω-values, that come from slightly too high values for the respective ratios. Thus, on average we get slightly positive $\tilde{\omega}$-values for the UDG-model $\omega = 0$.

On the other hand $\tilde{\omega}_{SD/E}$ and $\tilde{\omega}_{SD/M}$ are significantly underestimating large values of ω. An indicator for a large ω are very rare but very long edges. It requires an even larger sample size to appropriately cover those long edges. Thus, in the usual case of long edges not occurring, ω is slightly underestimated. In the rare case, that such a long edge exists it is over estimated, and rounding to $\tilde{\omega}_\bullet = 6.5$ causes a rounding error. Similar but reversed to the UDG-case this results in slightly underestimated ω-values on average. Long edges have a minor influence on E, a major influence on SD and almost no influence on M. That is why $\tilde{\omega}_{SD/E}$, which contains both affected statistics on average underestimates ω, and $\tilde{\omega}_{E/M}$, which does not contain the majorly influenced statistic SD, yields the most solid estimation for large ω.

Since ω usually gets under estimated, we fix the maximum value of the three options as ω-estimation:

$$\tilde{\omega} := max\{\tilde{\omega}_{SD/E}, \tilde{\omega}_{SD/M}, \tilde{\omega}_{E/M}\}$$

Because of the previous discussion, usually $\tilde{\omega}_{E/M}$ is the maximum.

Next we have a look at the three options $\tilde{r}_E$, $\tilde{r}_M$ and $\tilde{r}_{SD}$ to estimate the theoretical communication distance $r = 200$.

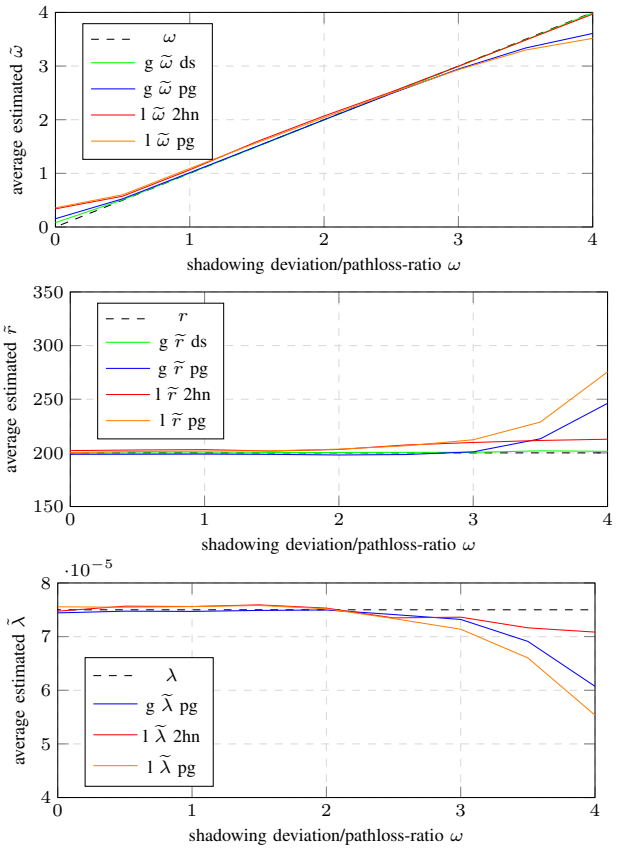


Fig. 2: Average estimated $\tilde{\omega}$, $\tilde{r}$ and $\tilde{\lambda}$ based on global (g) or local (l) information in the distance set (ds), 2-hop neighborhood (2hn) and proper graph (pg) simulations as well as the real values.

There is no explicit preference between the three options. It turns out, that using the medium value (i.e. the median of the three options) yields a solid estimation robust to outliers in either direction:

$$\tilde{r} := M[\{\tilde{r}_{SD}, \tilde{r}_E, \tilde{r}_M\}]$$

Since the sampled edge lengths cover no information about the degree of a graph, λ cannot be discussed at this point. As there is no choice when estimating the density, our estimation strategy can be summarized as follows:

- $\tilde{\omega} := max\{\tilde{\omega}_{SD/E}, \tilde{\omega}_{SD/M}, \tilde{\omega}_{E/M}\}$
- $\tilde{r} := M[\tilde{r}_E, \tilde{r}_M, \tilde{r}_{SD}]$
- $\tilde{\lambda} := \hat{N}/2\pi \int_0^\infty dp_{\tilde{r}}^{(\tilde{\omega})}(d)\mathrm{d}d$

2) 2-hop Neighborhood: In Figure 2 (red plots) we can see the average estimated parameters from the generated 2-hop neighborhood compared to the real values (dashed plots).

Some statistical effects are worth to be mentioned. The effect of rounding small ratios of $\hat{SD}$, $\hat{E}$ and $\hat{M}$ to $\omega = 0$, that can not be canceled out, resulting in overestimating positive $\tilde{\omega}$-estimations are still visible and due to smaller sample size are much more pronounced compared to the distance set simulation (green plots). For $\omega = 0$ the error increases to

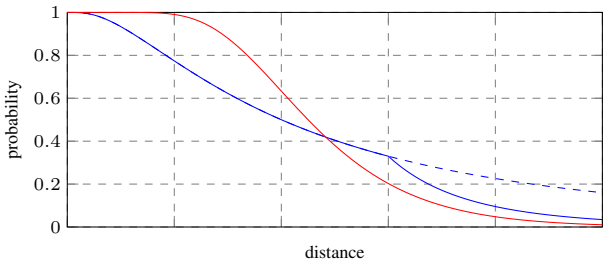


Fig. 3: Illustration of the boundary effect on the probability function. There are less detected long edges (blue) than there would be without the boundary (dashed). The estimation strategy compensates it with a lower ω and higher r (red).

0.34. The rounding error for small ω-values is visible up to $\omega = 2.5$.

Furthermore, the existence and nonexistence of very long edges are still noticeable. As explained, very long edges are very rare events (the longer the edge and the smaller ω, the rarer their appearance) and have a larger influence due to smaller sample sizes of the statistics. As a consequence, r gets over- and λ gets underestimated. This effect will be explained in detail in the following section, when boundary effects majorly cause the non-existence of long edges.

3) Proper Graph: As can be seen in Figure 2, the effects for large and small ω are visible again. Especially for large ω-values the real value seems to be heavily underestimated. The boundary of the simulation area cancels primarily long edges, which are an indicator for a high ω-value (see Section IV-B1). We can observe an inaccuracy resulting from that boundary effect. Deleting over-proportionally many of the long edges biases the measured distribution of edge lengths. As a consequence ω gets under- and r gets overestimated (see Fig.3).

Finally the density λ is estimated. It turns out to be quite precise for small ω. For $\omega > 2$ again long edges are more likely. Thus, the boundary cancels more of them and the observed number of neighbors is too low. Overestimating r intensifies this effect, since a higher $\tilde{r}$ suggests a higher average number of neighbors, than the graph actually has. Thus, the density is significantly underestimated due to boundary effects for $\omega > 3$.

In Figure 2 we can also observe that these trends combined with a smaller number of edges evaluated for a single estimation are intensified in the local estimation strategy.

It's worth to have a closer look at the regional differences of the estimated values in the graph, depicted in Figure 4. Note, that due to a smaller simulation area, boundary effects are much more intense.

For small ω-values we evaluate Figures 4a to 4c. We observe regions where ω is collectively underestimated. These regions, according to the previous considerations, respectively overestimate r. In the density-estimation these trends are not that sharp. However, there are regions with higher or lower estimated densities. All in all despite there are different regions estimating high or low values in the different parameters, they are spread over the whole simulation area.

For large ω-values instead the boundary effects dominate the scene. The distribution of edge lengths differs depending on the node being close to the boundary or roughly in the middle of the area. For a node right in the centre of the simulation area, the longest possible edge has length half of the diagonal of the simulation rectangle. On the other hand, for a node right in the corner of the simulation area, an edge covering the whole diagonal is now possible.

All in all, the closer a node is to the boundary, the more long edges it detects. Since these long edges are crucial to estimate high ω-values, the boundary effect of an underestimated ω (Figure 4d) and its consequential errors (Figures 4e, 4f) are surprisingly the less significant the closer one actually gets to the boundary.

V. Application

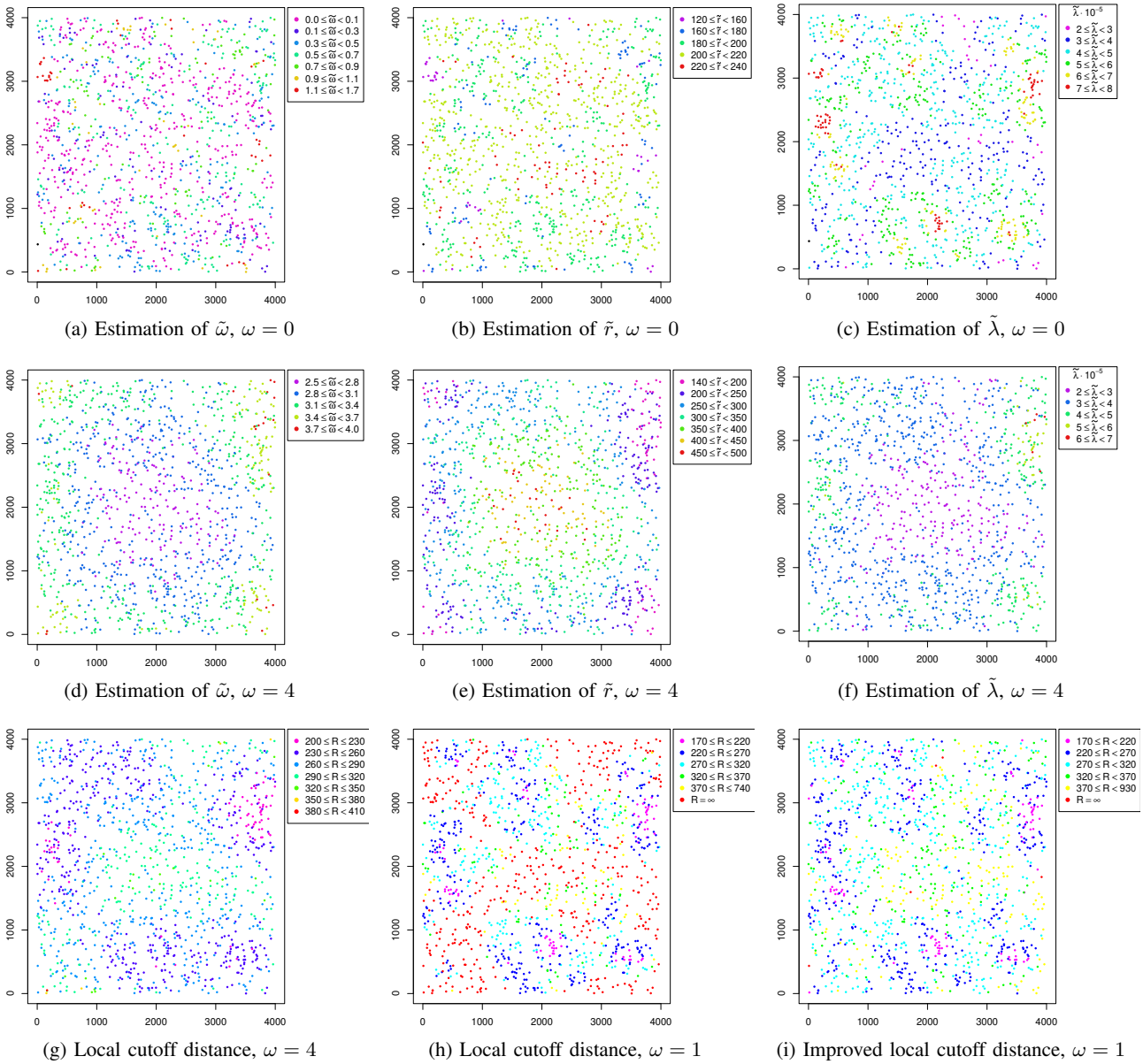
In [1] it was shown, that via applying cutoff R from (2) and subsequently the EAR-algorithm an almost connected and intersection free subgraph can be found. From over 1.3 billion intersections on average just about 70 have been left for the extreme case $\omega = 10$. For our largest tested $\omega = 4$ from over 5 million intersections on average about 17 have been left. Throughout all tested ω-values on average the giant component contained at least 99.3% of the nodes.

A. Simulation methodology

To compare the results to [1] we adapt the simulation area 4000×4000. The ω-scope 0 to 4 covers the majority of realistic σ- and α-values [4] and the theoretically interesting UDG-case $\omega = 0$. All other parameters remain the same as in the previous section. Like in [1] we sample 100 graphs for each ω. To compare the results independent of the respective sampled graphs we first run the procedure from [1], applying the adjusted global cutoff R and subsequently EAR, as well as applying the local cutoff and subsequently EAR. The difference between the two cutoffs is, that the global cutoffs are calculated with the original parameters, while the local one uses those from our locally applied estimation strategy. Note that the two end nodes of an edge can compute different cutoff distances due to different estimations. In this case the maximum of the two distances is chosen or in other words, an edge is deleted if and only if both endpoints agree on dropping it. We compare the size of the giant component after both procedures to that of the original graph, as well as the respective numbers of intersections.

B. Results of the Simulation

Consider first of all Figure 5, which depicts the average computed cutoff distance compared to the global cutoff distance from [1]. Note, that the average just considers those nodes, which were able to compute a cutoff distance. Nodes which estimated $\tilde{\omega} = 0$ or estimating a low $\tilde{\lambda}$, such that no solution to the equation in (2) exists, did not apply any cutoff. However, compared to the global cutoff some nodes were able to compute a cutoff in the UDG-regime $\omega = 0$ via estimating



(a) Estimation of $\tilde{\omega}$, $\omega = 0$ (b) Estimation of $\tilde{r}$, $\omega = 0$ (c) Estimation of $\tilde{\lambda}$, $\omega = 0$

(d) Estimation of $\tilde{\omega}$, $\omega = 4$ (e) Estimation of $\tilde{r}$, $\omega = 4$ (f) Estimation of $\tilde{\lambda}$, $\omega = 4$

(g) Local cutoff distance, $\omega = 4$ (h) Local cutoff distance, $\omega = 1$ (i) Improved local cutoff distance, $\omega = 1$

Fig. 4: Nodes colored according to their respective locally estimated parameter values and calculated cutoff distances.

$\tilde{\omega} > 0$, resulting in the average cutoff distance at $\omega = 0$ being finite in the plot.

For ω-values between 0.5 and 2.5, the average local cutoff distance approximates the global cutoff quite well. However, for $\omega \geq 3$, when the boundary effects arise and reduce the estimated $\tilde{\omega}$, the local cutoffs exceed the global cutoffs on average, since the scaling factor $(1+1/\omega))$ now is overestimated.

Next we compare the size of the giant component shown in Figure 6.

For small values of ω (1.0 and below) both cutoffs , global and local, do not affect the size of the giant component. That is due to small estimated $\tilde{\omega}$ and actual ω imply large scaling factors $(1 + 1/\omega)$ resp. $(1 + 1/\tilde{\omega})$. Combined with long edges to be very unlikely in LNS-graphs close to the UDG-model cutoff is not relevant. Recall, that the subsequent application of the EAR-algorithm does not change the connectedness.

For medium ω-values (1.5 to 3.0) the GC-size of the two cutoffs diverge. The global cutoff starts to matter and removes more and more large edges, that reduces the size of the giant component. The local cutoff, however, does not affect the GC. Here we can detect a major advantage of the local method. Since it just uses local information it adjusts to the local conditions of a graph. It removes longer edges in denser regions and keeps them where they are crucial.

For larger ω-values (3.5 and 4.0) the local cutoff also slightly effects the size of the giant component. However, it

	Original Graph		Global Cutoff		Local Cutoff			Improved Local Cutoff		
ω	$\|GC\|$	$\|IP\|$	$\|GC\|$	$\|IP\|$	$\|GC\|$	$\|IP\|$	$\|LCO\|$	$\|GC\|$	$\|IP\|$	$\|ILCO\|$
0.0	99.86	10,605	99.86	0.00	99.86	0.00	50.19	99.85	0.00	65.72
0.5	99.90	11,894	99.90	0.06	99.90	0.06	68.11	99.89	0.07	89.37
1.0	99.96	17,007	99.96	0.51	99.96	1.69	64.07	99.94	0.72	99.74
1.5	99.98	29,890	99.95	1.69	99.98	21.19	55.03	99.97	2.83	99.97
2.0	99.99	65,279	99.93	3.45	99.99	143.65	55.95	99.98	6.06	100.00
2.5	100.00	173,816	99.92	5.82	100.00	482.53	72.00	99.99	9.96	100.00
3.0	100.00	533,465	99.91	8.85	100.00	747.43	88.52	100.00	13.42	100.00
3.5	100.00	1,751,360	99.85	13.62	99.99	348.82	97.77	99.99	18.65	100.00
4.0	100.00	5,285,204	99.79	16.69	99.98	45.10	99.91	99.98	23.50	100.00

TABLE II: Relative size of the giant component ($|GC|$) and number of intersection points ($|IP|$) after applying global, local resp. improved local cutoff distance and subsequently EAR compared to the original graph; fraction of nodes, that were able to compute a local ($|LCO|$) resp. improved local cutoff($|ILCO|$).

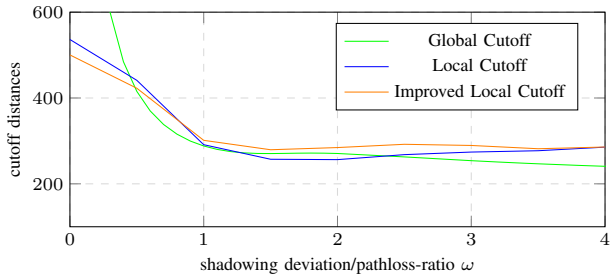


Fig. 5: Global, local and improved local cutoff distances.

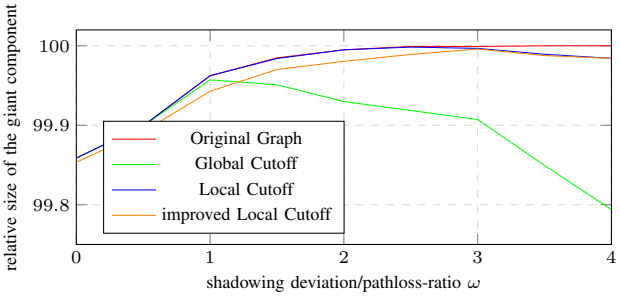


Fig. 6: Relative size of the giant component after applying global, local and improved Local cutoff.

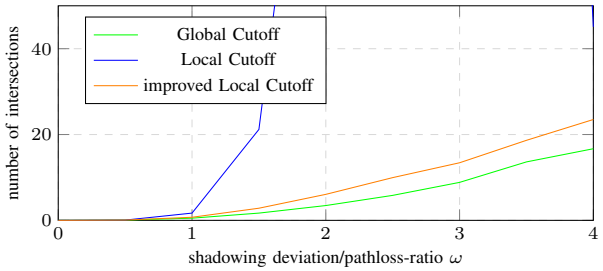


Fig. 7: Number of intersections after applying global, local and improved local cutoff and subsequently EAR.

does significantly less than the global cutoff.

To show how the regional variation of the parameter estimation affects the locally calculated cutoff distance, in Figure 4g, the nodes of an example graph are colored accordingly. The picture generally follows the previously mentioned trends from Figure 4. The limitation of the simulation area, which leads to an underestimated $\tilde{\omega}$, is compensated by higher cutoff distances for the central nodes. More dense regions and thus higher estimated values of $\tilde{\lambda}$ lead to shorter cutoff distances, since nodes with many close neighbors do not depend on long edges.

Finally, we take a look at the number of intersections left after the application of EAR (Figure 7). Here we can detect the major weakness of the local cutoff. For small ω-values (0.0 and 0.5), where the different cutoffs have the same effect on the edges (in particular almost none), the number of intersections is the same. However, beyond $\omega = 1$ the global cutoff significantly outperforms the local one. This peaks in ω-values 2.5 and 3.0 yielding more than 80 times as many intersections in the local version compared to the global version (Table II). This weakness will be addressed in the next section.

C. Improving the local cutoff

In this section we address the intersection number peak occurring in Table II. These intersections are mainly generated by long edges, where the local cutoff calculation fails, and hence they are not canceled before the application of the EAR-algorithm. To understand where this failure comes from we take a closer look at the calculation of R.

The task is to find the smallest x to solve the equation from (2): $p(x)x^2 = {}^{1.437}/_{\lambda}$. Since for the LNS-model $\lim_{x\to\infty} p(x)x^2 = 0 = p(0)0^2$, there exists a solution iff. $\max\{p(x)x^2\} \geq {}^{1.437}/_{\lambda}$. This maximum value increases with increasing ω. So for small ω an underestimated λ more likely causes the failure of calculating any cutoff distance. That can be seen in Table II column $|LCO|$, which yields the number of nodes that are able to calculate a local cutoff distance.

On the other hand for small ω-values the scaling factor $(1+{}^1/_{\tilde{\omega}})$ and hence the eventual cutoff is quite large and long edges are very rare. Thus, practically no cutoff is enforced anyway with high probability.

In conclusion, for large ω-values it is not very likely that the cutoff distance calculation fails, while for small ω-values missing cutoff distances do not cause a problem in the application of EAR. So, these intersections caused by single long edges that are not deleted in the cutoff step occur only for the intermediate ω-values.

The idea to fix this is based on the tendency to underestimate λ (see section IV-B3).

Assuming that we underestimated λ, we increase the density until we find a solution to (2). Independent of the λ when the first solution occurs, the corresponding infimum will be $\operatorname{argmax}_x\{p(x)x^2\}$. So when we cannot find a proper solution, we assume that the used $\tilde{\lambda}$ is underestimating the density and we use the maximum point as pretended infimum.

Note, that by the scaling factor $(1 + {}^1/_{\omega})$ the calculated infimum is scaled the more the smaller ω gets and the cutoff calculation mainly fails for small ω-values. So the

improved local cutoff is usually safe above the theoretical communication distance $r = 200m$ and edges exceeding it are very unlikely for small ω. It does not affect the size of the giant component significantly in our simulations.

D. Results with the Improved Cutoff

Finally, we evaluate the performance of the improved local cutoff.

In Table II we see that now just for the ω-values 0.0 and 0.5 the fraction of nodes not calculating a cutoff is notable. Isolated nodes are statistically not relevant. And furthermore an eventual cut off distance would not have any effect. The only remaining possibility for non-isolated nodes to not obtain a cutoff distance is when the estimated $\tilde{\omega}$ is indeed 0 and hence the scaling factor $(1 + {}^1/\tilde{\omega})$ is not defined.

In Figure 5 we see that for these two ω-values the applied cutoffs are above 400. Since in this regime edges even exceeding $r = 200$ are not possible ($\omega = 0$) or very unlikely ($\omega = 0.5$) there is technically no cutoff enforced by any node. For all other ω-values the now by almost all nodes calculated improved local cutoff is on average higher than both the local and global cutoff. Thus, the choice of the maximum point to calculate the improved local cutoff is justified, and does not seem to tackle the connectedness.

Taking a closer look on an example graph in Figure 4 further confirms that the values for the improved cutoff distance are reasonable. The red colored nodes in Figure 4h could not calculate a local cutoff distance, so the cutoff distance is set to ∞. The colorization according to the improved cutoff distance in Figure 4i however, shows that those nodes who can now calculate a cutoff distance obtain similar values as their neighbors do.

Note, that every node keeps its original local cutoff, if there was one, and only the nodes who newly can calculate a cutoff reduce them from ∞ to a real number. That is why the improved local cutoff deletes at least all edges the local cutoff has deleted before and the giant component can not be larger than before. However, it isn't significantly smaller either. Indeed for $\omega > 1$ the improved local cutoff still outperforms the global cutoff in terms of giant component size (Figure 6).

In terms of intersection numbers (Figure 7) the improved local cutoff is a huge improvement compared to the local cutoff and gets very close to the global cutoff performance.

Summarizing, although the improved local cutoff just processes local data, and hence provides a purely local and distributed planarization scheme, it yields results, that are quite comparable to the global cutoff variant.

VI. Conclusion

We presented a new way to estimate global key parameters of the log-normal shadowing model along with the node density of a network that can be applied fully local by each node individually. In various simulations we showed that the estimation strategy closely predicts the original values. Just boundary effects resulted in estimations significantly diverging from the original parameters. The observed inaccuracy for parameters close to the UDG-model is not relevant for a real usage scenario since those parameters are unlikely to occur in practice.

Furthermore, we showed via simulations that the local planarization strategy presented in [1] (which is depending on global parameters) can be refined to a fully localized strategy by using the locally estimated parameters. The estimation approach is suitable to be combined with the EAR-algorithm, since both require edge length information only. Our simulations showed that this solution yields results very close to the original strategy, but without any knowledge of global parameters. It turned out, that local estimation inaccuracies were beneficial to compensate local topology anomalies like particularly sparse and dense regions or network boundaries.

Further research on node distributions other than a homogeneous PPP and regionally varying LNS-parameters, especially from the perspective of the local adaptability described above, as well as examining different applications relying on the network parameters and transfer to the $\mathbb{R}^3$-space are subject to future work.

References

[1] S. Böhmer, L. Böltz, and H. Frey, "Wip: Local heuristics for very likely connected and intersection free wireless network topologies under log-normal shadowing," in *2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 165–168, 2022.

[2] S. Böhmer, D. Schneider, and H. Frey, "Stochastic modeling and simulation for redundancy and coexistence in graphs resulting from log-normal shadowing," in *Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*, p. 173–182, 2019.

[3] C. Bettstetter and C. Hartmann, "Connectivity of wireless multihop networks in a shadow fading environment," in *Proceedings of the 6th ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*, p. 28–32, 2003.

[4] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Prentice Hall PTR, 1996.

[5] H. F. Ates, S. M. Hashir, T. Baykas, and B. K. Gunturk, "Path loss exponent and shadowing factor prediction from satellite images using deep learning," *IEEE Access*, vol. 7, pp. 101366–101375, 2019.

[6] H. Nurminen, J. Talvitie, S. Ali-Löytty, P. Müller, E.-S. Lohan, R. Piché, and M. Renfors, "Statistical path loss parameter estimation and positioning using rss measurements in indoor wireless networks," in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–9, 2012.

[7] S. Srinivasa and M. Haenggi, "Path loss exponent estimation in large wireless networks," in *2009 Information Theory and Applications Workshop*, pp. 124–129, 2009.

[8] Y. Hu and G. Leus, "Self-estimation of path-loss exponent in wireless networks and applications," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 11, pp. 5091–5102, 2015.

[9] G. Mao, B. D. Anderson, and B. Fidan, "Path loss exponent estimation for wireless sensor network localization," *Computer Networks*, vol. 51, no. 10, pp. 2467–2483, 2007.

[10] S. Böhmer, L. Böltz, and H. Frey, "Local construction of connected and plane spanning subgraphs under acyclic redundancy," in *2020 18th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*, pp. 1–8, 2020.