Constrained Localization: Mapping Wireless Sensor Nodes in Predefined Positions

Andrea Bardella*, Nicola Bui*†, Andrea Zanella* and Michele Zorzi*†
*Department of Information Engineering, University of Padova, Padova, Italy – Emails: {bardella, bui, zanella, zorzi}@dei.unipd.it
†Patavina Technologies, Padova, Italy – Email: {nicola.bui, michele.zorzi}@patavinatech.com

***Abstract*—This paper proposes a novel method for solving localization problems leveraging on node position constraints. This consists in mapping n wireless nodes onto n predefined positions in a map. The problem may be solved by first applying standard localization algorithms to get an initial estimate of the node positions in the area and, successively, mapping each estimated position to the closest admissible point in the map. Results can be improved by applying algorithms that are explicitly designed to manage the available information for the constrained problem. In this study, we propose three algorithms, based on a greedy, multi dimensional scaling, and belief propagation approach, respectively. The algorithms are analyzed and compared by using synthetic data. Results reveal that the belief propagation approach, suitably modified to account for the position constraints, outperforms the other algorithms in all the considered settings.**

I. Introduction

Localization in wireless sensor networks (WSNs) is a challenging problem that has generated a large body of literature [1]–[3]. Typically, localization algorithms encompass the presence of a limited number of nodes, called *anchors* (also known as beacons or landmarks), which are aware of their geographical positions. Anchors are required to periodically broadcast their coordinates in order to let the other nodes in the network, referred to as *strayed* nodes, infer their own position by means of some estimation technique.

Although the localization problem has been considered in many different scenarios (indoor and outdoor, static and mobile, and so on), a common assumption is that strayed nodes can occupy *any* position in the area. This assumption reflects the vision of randomly deployed WSNs, where nodes are scattered in an area without any predetermined scheme. However, in several practical scenarios, nodes are placed according to predefined patterns that satisfy application-specific requirements. For example, sensors can be placed on the regular slots of a large parking lot to monitor vehicle movements and parking occupancy, or attached to the works of art in a museum as anti-theft devices, or planted into the terrain, forming regular patterns, to monitor the soil conditions in greenhouses or fields under cultivation. In these scenarios, the set of positions is known beforehand, and what needs to be determined is a bijective function associating nodes and positions.

To the best of our knowledge, this is the first attempt to add stringent *position constraints* to the classic localization problem. Exhaustive search approaches can be used when nodes are few, but they become soon infeasible as the size of the problem grows. Another intuitive approach is to first get an unconstrained localization of the nodes, by means of standard localization algorithms, and successively map each node to the closest predefined point. However, when the localization error of the unconstrained algorithm is comparable with the mean internode distance, the mapping into the predefined positions may easily swap close-by nodes.

In this paper, we investigate the constrained localization problem by proposing three different algorithms, called *Greedy* (GREED), *Adapted Multi-Dimensional Scaling* (AMDS), and *Constrained Belief Propagation* (CBP). The GREED algorithm iteratively assigns nodes to the predefined positions based on of a simple greedy strategy. The AMDS algorithm, instead, first solves the unconstrained localization problem by using the multidimensional scaling localization algorithm, and then maps the nodes back to the predefined points in the map. Finally, CBP is a revisitation of the belief propagation scheme applied to a graph with loops.

The rest of the paper is organized as follows. Sec. II reviews the literature; the system model is defined in Sec. III; the three localization algorithms are described in Sec. IV, whereas Sec. V presents and discusses our results. Sec. VI concludes the paper.

II. Related Work

Our review focuses on range–based algorithms, which make use of physical measurements to estimate the distance between strayed (non-localized) nodes and anchor (position-aware) nodes. The Received Signal Strength Indication (RSSI) is often used for distance estimation, since it is natively supported by many devices. However, RSSI-based ranging is usually rather poor due to the

stochastic nature of radio propagation [4], [5]. The reader can refer to [6]–[8] for a more complete overview of these techniques.

Reference [3] suggests that location estimation accuracy can be improved by adding ranging data obtained by strayed nodes. A practical algorithm that makes use of pairwise inter-node range estimates is proposed in [9], where the authors apply the Multi-Dimensional Scaling (MDS) method to solve the localization problem. However, this model does not directly apply, since in our scenario positions lie on a finite lattice.

Other approaches can be found in [10], [11], where factor-graph and nonparametric belief propagation approaches are used. Again, their system models cannot directly apply to our scenario. Nonetheless, the belief propagation approach is very attractive, due to its capability of capturing and managing inter-node interactions in an effective manner.

III. System model

We consider a network with N nodes, out of which $m < N$ are anchor and $n = N - m$ are strayed nodes. Nodes are placed in N positions on the map. We denote by A and S the sets of identifiers of the anchor and strayed nodes, respectively, and by $\Phi = \{\varphi_i\}_{i=1}^{N}$ the set of all admissible positions. Each node occupies a single position, in an exclusive manner. Let $\mathbf{X} = [X_1, \ldots, X_N]$ denote the vector of node positions. Anchor node positions are known beforehand ($\Pr[X_i = x_i] = 1,\ i \in A$). Thus, let $\Phi_A = \{x_i\}_{i \in A}$ and $\Phi_S = \Phi \setminus \Phi_A$ be the sets of the positions of the anchor and the strayed nodes, respectively. Since the position of strayed nodes is initially unknown, $\Pr[X_j = \varphi_k] = 1/n,\ j \in S,\ \varphi_k \in \Phi_S$. Note that the random variables $\{X_j\}_{j \in S}$ are identically distributed, but not mutually independent because of the *exclusivity condition*, according to which any position can be occupied by a single node ($\Pr[X_i = X_j] = 0,\ i, j = 1, \ldots, N,\ i \neq j$).

Let $\Gamma_{i,j}$ be the RSSI value for the link between nodes i and j. We assume by reciprocity that $\Gamma_{i,j} = \Gamma_{j,i}$. Although this condition is not strictly verified in practice, we can always consider $\Gamma_{i,j}$ as the average of a series of RSSI samples collected separately by nodes i and j over a certain time interval. In this way, furthermore, we also average out the time fluctuations of the received signal power due to fast fading and mitigate hardware calibration errors.

According to [12], the power received at distance d from the transmitter can be roughly expressed as

$$\Gamma_{i,j} = f(d) + \Psi_{i,j}\,, \tag{1}$$

where $f(d) = P_{tx} + K - 10\eta \log_{10}\left(\frac{d}{d_0}\right)$ is the deterministic path loss function, P_{tx} is the transmit power [dBm], K is the nominal received power at the reference distance d_0, η is the path loss coefficient and $\Psi_{i,j}$ is a random variable, with density $p_\Psi(\cdot)$, that accounts for the shadow fading. Typically, $\Psi_{i,j}$ is modeled as a zero-mean Gaussian random variable, with standard deviation σ_Ψ that ranges from 2 to 6 dB, depending on the environment. In this paper, however, we assume that $\Psi_{i,j}$ has a Gumbel distribution, which better fits the experimental data collected in our testbed [13]. Although the shadowing in general may present spatial correlation, for simplicity, we assume it iid in this study.

Thus the PDF of $\Gamma_{i,j}$ over a distance d can be expressed as

$$p_\Gamma(\gamma_{i,j}|d) = p_\Psi(\psi_{i,j}) \tag{2}$$

where $\psi_{i,j} = \gamma_{i,j} - f(d)$. The estimate $\hat{d}$ of the distance between the nodes, given the RSSI value $\gamma_{i,j}$, is given by

$$\hat{d}(\gamma_{i,j}) = f^{-1}(\gamma_{i,j}) = d_0 10^{\frac{P_{tx}+K-\gamma_{i,j}}{10\eta}} = d10^{\frac{\psi_{i,j}}{10\eta}}\,, \tag{3}$$

which shows that the ranging error is multiplicative with the distance. For $d < d_0$ the path loss propagation model (1) does not hold, so that the accuracy of (3) is limited.

The aim of the constrained localization algorithms is to determine the most likely node position vector $\mathbf{x} = [x_1, \ldots, x_N]$, given the (symmetric) matrix of RSSI observations $\mathbf{g} = [\{\gamma_{i,j}\}_{i,j}]$. Then, denoting by $\mathbf{\Phi} = [\varphi_1, \ldots \varphi_N]$ a vector of positions in Φ, we have

$$\mathbf{x} = \arg\max_{\mathbf{\Phi}} \Pr[\mathbf{X} = \mathbf{\Phi} | \mathbf{\Gamma} = \mathbf{g}] \tag{4}$$

where the conditional joint probability can be expressed as

$$\begin{aligned} &\Pr[\mathbf{X} = \mathbf{\Phi} | \mathbf{\Gamma} = \mathbf{g}] = \qquad (5)\\ &= \frac{\Pr\left[\{\Gamma_{i,j} = \gamma_{i,j}\} \,\middle|\, \{X_i = \varphi_h\}\right] \cdot \Pr[\{X_i = \varphi_h\}]}{\Pr[\{\Gamma_{i,j} = \gamma_{i,j}\}]} \propto \\ &\propto \prod_{i \in S} \prod_{\substack{j \in A \cup S \\ j \neq i}} p_\Gamma(\gamma_{i,j} \,|\, ||\varphi_h - x_j||) \prod_{i \in A \cup S} \Pr[X_i = \varphi_h]\ ; \end{aligned}$$

where $||\varphi_h - x_j||$ is the Euclidean distance between the positions φ_h and x_j associated with node i and j, respectively. The second and third lines of (5) follow from the Bayes rule and from the independence assumption of the shadowing terms in (1), whereas $\Pr[\{\Gamma_{i,j} = \gamma_{i,j}\}]$ is constant since $\gamma_{i,j}$ are observations.

IV. Constrained localization algorithms

The maximization problem (4) could be solved by checking the likelihood of each possible permutation of the n positions in Φ_S ($n!$ possible solutions), but the this

procedure is infeasible for n greater than a few units. In the following we propose three algorithms that yield suboptimal results in polynomial time.

A. Greedy algorithm (GREED)

The design of the GREED algorithm favors the low computational complexity over the quality of the results. The algorithm performs n successive steps *deciding* one strayed node at a time, i.e., permanently assigning it a position according to a greedy criterion described hereafter.

Let $D^{(\ell)}$ and $U^{(\ell)}$ denote the sets of decided and still undecided nodes at the beginning of step $\ell = 1, \dots, n$, with $D^{(1)} = A$ and $U^{(1)} = S$. Similarly, let $\Phi_D^{(\ell)}$ be the set of positions assigned to decided nodes, and $\Phi_U^{(\ell)}$ the set of still available positions, with $\Phi_D^{(1)} = \Phi_A$ and $\Phi_U^{(1)} = \Phi_S$.

For each undecided node $i \in U^{(\ell)}$ and each $h \in \Phi_U$, we compute the conditional probability that $X_i = \varphi_h$, given the vector $\{\gamma_{i,j}\}_{j \in D^{(\ell)}}$ of RSSI values, as

$$\Pr\left[X_i = \varphi_h \middle| \Gamma_{i,j} = \gamma_{i,j}\,,\ \forall j \in D^{(\ell)}\right] = \tag{6}$$
$$= \prod_{j \in D^{(\ell)}} \frac{p_\Gamma(\gamma_{i,j} \,|\, ||\varphi_h - x_j||) \Pr\left[X_i = \varphi_h\right]}{\Pr\left[\Gamma_{i,j} = \gamma_{i,j}\right]}$$

where $||\varphi_h - x_j||$ is the Euclidean distance between positions φ_h and x_j, and x_j is the position permanently assigned to node j. The second line of (6) follows from the Bayes rule and the independence of the shadowing terms in (1).

We observe that, since the *a priori* distribution of X_i over the available positions is uniform, the conditional probabilities (6) for any $i \in U^{(\ell)}$ and $\varphi_h \in \Phi_U^{(\ell)}$ are proportional to the coefficients

$$P_{i,h} = \prod_{j \in D^{(\ell)}} p_\Psi(\gamma_{i,j} - f(||\varphi_h - x_j||))\,, \tag{7}$$

where we used (2). These coefficients are collected in a square matrix $\mathbf{P}$ of size $n - \ell + 1$, whose rows are associated to the nodes in $U^{(\ell)}$, and columns to the positions in $\Phi_U^{(\ell)}$.

Considering that the association between nodes and positions is bijective, the matrix $\mathbf{P}$ should be doubly stochastic. Unfortunately, this is not always the case, since the conditional probabilities $P_{i,h}$ are computed independently for each node i. However, by alternatively normalizing the rows and columns of $\mathbf{P}$, according to the *Sinkhorn scaling process* [14] and under the condition that the matrix is *scalable*, we obtain a sequence of matrices that converges to a unique doubly stochastic matrix $\bar{\mathbf{P}}$.

From $\bar{\mathbf{P}}$, we then select the largest term $\bar{P}_{i,h}$ and we permanently assign node i to position φ_h, by setting $x_i = \varphi_h$. The state variables are finally updated as follows:

$$D^{(\ell+1)} = D^{(\ell)} \cup \{i\}\,; \quad U^{(\ell+1)} = U^{(\ell)} \setminus \{i\}\,;$$
$$\Phi_D^{(\ell+1)} = \Phi_D^{(\ell)} \cup \{\varphi_h\}\,; \quad \Phi_U^{(\ell+1)} = \Phi_U^{(\ell)} \setminus \{\varphi_h\}\,. \tag{8}$$

The algorithm ends after exactly n steps, when each strayed node has been permanently associated to one position.

B. Adapted Multidimensional Scaling (AMDS)

The Adapted Multidimensional Scaling (AMDS) algorithm simply consists in interlacing the execution of the original MDS algorithm with a remapping phase that assigns each node to the closest admissible position. For space constraints, we omit the details of the MDS algorithm, for which we refer the reader to [9], and describe here only its main components. As for GREED, the algorithm works in consecutive steps, deciding the permanent association between a strayed node and an admissible position at each step, and then using the decided nodes as anchors in the subsequent steps. We can thus inherit the notation introduced in Sec. IV-A to indicate which nodes and positions are decided, and which are still undecided at the beginning of the ℓth step.

At every step, the MDS algorithm is run on the instance of the problem where nodes in $D^{(\ell)}$ are considered anchors, and nodes in $U^{(\ell)}$ are strayed. Note that the MDS algorithm is also iterative so that each step of AMDS actually requires several iterations of the MDS algorithm.

Let $\mathbf{X}^{(k)} = [X_1^{(k)}, \dots, X_N^{(k)}]$ be the vector of positions assigned to the N nodes, at the kth iteration of the MDS algorithm. Positions corresponding to decided nodes are permanent, i.e., $X_j^{(k)} = x_j$ for each $j \in D^{(\ell)}$. Conversely, positions of undecided nodes are *temporary* and may change during the execution of the MDS algorithm. The position of each undecided node $i \in U^{(\ell)}$ is initialized by considering the coefficients $P_{i,h}$ defined in (7), and selecting the position $\varphi_h \in \Phi_U^{(\ell)}$ for which $P_{i,h}$ is maximum.

Given the position vector $\mathbf{X}^{(k)}$, the MDS algorithm computes the following cost function [9]

$$S(\mathbf{X}^{(k)}) = \sum_{i \in U^{(\ell)}} \sum_{\substack{j=1 \\ j \neq i}}^{N} \left[w_{i,j} \left(\hat{d}(\gamma_{i,j}) - ||x_i^{(k)} - x_j^{(k)}|| \right)^2 \right] \tag{9}$$

where $\hat{d}(\gamma_{i,j})$ is given in (3) and its accuracy is weighted by the coefficient $w_{i,j}$. $w_{i,j} = \alpha e^{-\gamma_{i,j}^2 / P_{th}^2}, \gamma_{i,j} \geq P_{th}$, with $\alpha = 1$ for $j \in A^{(\ell)}$, and $\alpha = 2$ for $j \in U^{(\ell)}$;

instead, $w_{i,j} = 0$ for $\gamma_{i,j} < P_{th}$. The parameter P_{th} is the power threshold below which the RSSI samples are not used for ranging and usually it coincides with the minimum received power to correctly decode incoming packets.

At each iteration, the MDS algorithm adjusts the positions of undecided nodes to minimize the cost function (9). Note that, in this phase, node positions are not constrained to the set of admissible points.

At the kth iteration, the temporary position of each undecided node $i \in U^{(\ell)}$ is updated as follows

$$x_i^{(k+1)} = a_i \mathbf{X}^{(k)} \mathbf{c}_i^{(k)} \tag{10}$$

where $a_i = \left(\sum_{j \neq i} w_{i,j}\right)^{-1}$ and $\mathbf{c}_i^{(k)} = \left[c_{i,1}^{(k)}, \ldots, c_{i,N}^{(k)}\right]^T$ is a column vector with elements

$$c_{i,j}^{(k)} = \begin{cases} w_{i,j}\left(1 - \dfrac{\hat{d}(\gamma_{i,j})}{||x_i^{(k)} - x_j^{(k)}||}\right), & j \neq i\,; \\ \displaystyle\sum_{s \neq i} w_{i,s} \frac{\hat{d}(\gamma_{i,s})}{||x_i^{(k)} - x_s^{(k)}||}, & j = i\,. \end{cases} \tag{11}$$

For decided nodes $j \in D^{(\ell)}$, we instead have $X_j^{(k+1)} = x_j^{(k)}$.

The step is completed at the first iteration t such that $S(\mathbf{X}^{(t-1)}) - S(\mathbf{X}^{(t)}) < \varepsilon$ for a certain $\varepsilon > 0$. At this point, the AMDS algorithm selects the node $i \in U^{(\ell)}$ that has been moved the least during the step, i.e.,

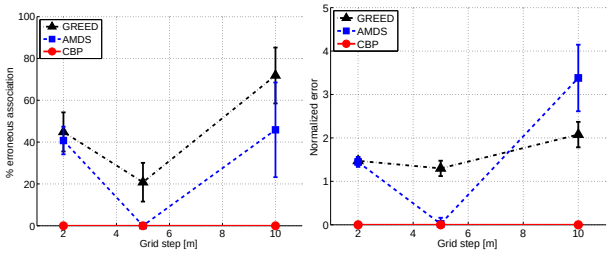
$$i = \arg \min_{h \in U^{(\ell)}} \{||x_h^{(1)} - x_h^{(t)}||\}\,.$$

Node i is then permanently assigned to the position $x_i = \varphi_h$, where $\varphi_h \in \Phi_U^{(\ell)}$ is at minimum distance from $\varphi_h^{(t)}$. The state variables are updated according to (8) and the algorithm proceeds with the following step, until all nodes are decided.

C. Constrained Belief Propagation (CBP)

The CBP localization algorithm, inspired by the Factor Graph theory [15] to which the reader is referred for an in-depth discussion of the subject, attempts to solve the optimization problem (4) by using a statistical approach [10], [11].

Basically, if the joint probability function that describes the problem can be factorized, such factorization can be represented with a graph. If the resulting graph is acyclic the calculation of the marginals or of the maximization problem turns out to be exact exploiting a message passing technique. Instead, if the graph has cycles, it is possible to look for a solution iterating the original procedure as long as convergence is achieved. However there is no guarantee that the method converges and, even



(a) Percentage of mismatches (b) Normalized misplacement error

Figure 1. Grid topology with $\sigma_\Psi = 2$ dB.

if it does, the solution might not be the optimum one. Notwithstanding these drawbacks, in several scenarios [16] this iterative approach is shown to achieve very good results [17], [18]. Moreover, the joint probability function might be handled to obtain a tractable graph either by unwrapping it [19] or by means of a generalized belief propagation junction tree approach [20] in order to converge to a meaningful solution.

Inspired by these works, we adapted the loopy belief propagation algorithm so that all system information provided by the measurements and the network's constraints is accounted for.

For each node i, the algorithm defines a *belief* function $b_i^{(\ell)}(\varphi_h), \varphi_h \in \Phi$, that is proportional to the probability that $X_i = \varphi_h$. The belief functions are initialized to the a priori distribution of X_i as discussed in Sec. III, i.e., $b_i^{(0)}(\varphi_h) = \Pr[X_i = \varphi_h]$. In the subsequent steps, the belief function of each node is updated taking into account the beliefs of the other nodes and the RSSI-based range estimates.

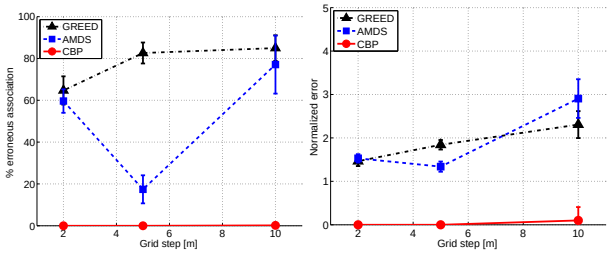
To this end, for each tuple of indices $\{i, j, h, k\}$, with $i \neq j$, we define the coefficients

$$\Theta_{i,j}(h,k) = \begin{cases} p_\Psi(\gamma_{i,j} - f(||\varphi_h - \varphi_k||))\,, & \gamma_{i,j} \geq P_{th},\ h \neq k; \\ F_\Psi(P_{th} - f(||\varphi_h - \varphi_k||))\,, & \gamma_{i,j} < P_{th},\ h \neq k; \\ 0\,, & h = k\,; \end{cases} \tag{12}$$

where $F_\Psi(\cdot)$ is the cumulative distribution function of Ψ. The coefficients $\{\Theta_{i,j}(h,k)\}_{i \neq j}$ are proportional to the joint probability that $X_i = \varphi_h$ and $X_j = \varphi_k$, given the RSSI observations $\{\gamma_{i,j}\}_{i,j}$. The last row in (12) accounts for the *exclusivity condition*, i.e., a position can not be occupied by more than one node.

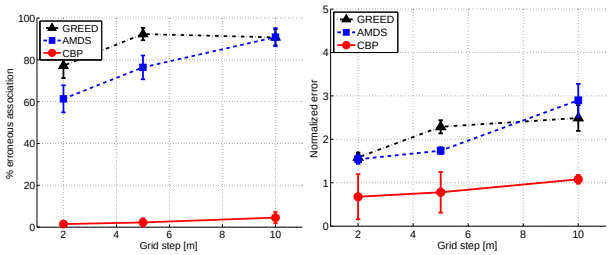
Then, at the ℓth iteration, the CBP algorithm computes for each node $i \in U$ and all other nodes $j = 1, \ldots, N$, $j \neq i$, the vector

$$\mu_{j \to i}^{(\ell)}(\varphi_h) \propto \left(\sum_{k=1}^{N} \Theta_{ij}(h,k) \frac{b_j^{(\ell-1)}(\varphi_k)}{\mu_{i \to j}^{(\ell-1)}(\varphi_k)}\right) \tag{13}$$



(a) Percentage of mismatches (b) Normalized misplacement error

Figure 2. Grid topology with $\sigma_\Psi = 4$ dB.



(a) Percentage of mismatches (b) Normalized misplacement error

Figure 3. Grid topology with $\sigma_\Psi = 6$ dB.

where proportionality is used as we normalize so that $\sum_{h=1}^{N} \mu_{j\to i}^{(\ell)}(\varphi_h) = 1$. Roughly speaking, $\mu_{j\to i}^{(\ell)}(\varphi_h)$ is a measure of belief that $X_i = \varphi_h$ from the perspective of node j. These vectors are then used to compute the pseudo-belief function for each node i as

$$\beta_i^{(\ell)}(\varphi_h) = b_i^{(\ell-1)}(\varphi_h) \prod_{j \neq i} \mu_{j\to i}^{(\ell)}(\varphi_h)\,, \qquad (14)$$

which is a modification of the original iterative belief propagation update rule obtained substituting the a priori probability with the updated one in order to *push* the beliefs until they converge to stable vectors with a single 1 and all 0s. Finally, in order to avoid the convergence to infeasible nodes' configurations, the Sinkhorn scaling process [14] is applied to the matrix $\mathbf{B}^{(\ell)}$ that collects the pseudo-belief $\beta_i^{(\ell)}(\varphi_h)$ for $i, h = 1, \ldots, N$, in order to obtain the doubly stochastic matrix $\hat{\mathbf{B}}^{(\ell)}$ whose elements $b_i^{(\ell)}(\varphi_h)$ are the updated belief functions. The CBP algorithm ends at step t when $\hat{\mathbf{B}}^{(t)}$ is a doubly stochastic matrix with only 0s and a single 1 element per row that univocally defines the position assigned to each node.

V. Performance analysis

Here, we will report the results obtained on synthetic scenarios, where ranging measures are artificially generated from (3). The scenarios consist of an 11 x 11 nodes grid, with $m = 9$ anchors and $n = 112$ strayed nodes. Anchor nodes are placed in the vertices, in the midpoints of each side and in the center of the grid. We fixed the parameters K and η of the deterministic path loss model, and varied the standard deviation of the shadowing term σ_Ψ from 2 to 4, and 6 dB, which corresponds to increasingly less reliable range estimates. The power threshold P_{th} was set in order to neglect RSSI samples from nodes farther than $R_{th} = 30$ meters, on average. The grid step d_{step} has been varied to 2, 5 and 10 meters, corresponding to three different values of the grid area V and, in turn, of the node density $\delta = N/V$. For each setting, we generated 50 realizations of the RSSI matrix, which have been used in all the localization algorithms, for the sake of a fair comparison.

Fig. 1 reports results obtained by the three algorithms for $\sigma_\Psi = 2$ dB varying the grid step. The leftmost graph shows the fraction of misplaced nodes, whereas the rightmost plot gives the mean positioning error of misplaced nodes, normalized to the grid step. Fig. 2 and Fig. 3 show the same metrics for $\sigma_\Psi = 4$ and 6 dB, respectively. Vertical bars represent the 95% confidence intervals.

All the algorithms degrade for increasing σ_Ψ, as the ranging becomes more noisy. For low σ_Ψ, AMDS and GREED exhibit a non-monotonic performance dependence on the grid step, with a step value that minimizes the fraction of misplaced nodes. Conversely, CBP shows a monotonic performance loss when the network density decreases, though it always outperforms the other algorithms misplacing a few nodes only in the harshest condition. Comparing the graphs on the right-hand side, we note that the misplaced nodes are usually within three grid steps from their actual position for AMDS and GREED, while CBP misplacement is limited to one grid step.

A final remark concerns the computational complexity of the three algorithms. According to our empirical results we can state that the execution of GREED is much faster than AMDS that, in turn, is quicker than CBP. Unfortunately, the iterative nature of AMDS and CBP does not allow an exact characterization of the algorithms complexity in terms of number of basic operations. Nonetheless, it is possible to get an idea of the asymptotic complexity of the algorithms by representing the number of iterations as a generic function $I(n)$, where n is the size of the problem. First of all, we observe that GREED performs n steps. At the generic ℓth step, GREED performs $m + \ell$ operations for $n - \ell$ nodes and $n - \ell$ positions, i.e., order of n^3 operations per step. Additionally, at the end of each step, the GREED algorithm executes the Sinkhorn scaling algorithm [14] to double-normalized the matrix of $n - \ell$ elements,

which requires $C_{ds}(n-\ell) = O((n-\ell)^3)$ operations [21]. Thus the overall complexity of the GREED algorithm is not greater than $O(n^4)$ operations. Similarly, the AMDS algorithm performs n steps. The ℓth step requires $(n-\ell)(n+m)^2$ operations for determining $c_{i,j}$, which are repeated for $I_{AMDS}(n-\ell)$ iterations to reach converge. Overall, then, the complexity of AMDS is $O(n^4 I_{AMDS}(n))$. Finally, the CBP algorithm requires n operations for computing the terms in (13), which are in the order of n^3, and other n operations for computing the n^2 elements (14). Furthermore, each iteration is completed by the execution of the Sinkhorn scaling algorithm. Therefore, each iteration of CBP takes a number of operations in the order of $n^4 + C_{ds}(n)$, so that, the overall complexity of the algorithm is $O(I_{CBP}(n)(n^4 + C_{ds}(n)))$. We observe that, assuming that $I_{AMBDS}(n)$ and $I_{CBP}(n)$ are of the same order, the asymptotic complexity of the AMDS and CBP is equal. However, the number of operations performed by CBP at each iteration remains constant, whereas AMDS progressively reduces the size of the problem at each step. Hence, in practice CBP takes more time than AMDS.

VI. Conclusion

In this work we addressed the problem of mapping n nodes to as many predefined positions. We proposed three algorithms of increasing complexity, namely GREED, AMDS and CBP. The simulations performed on a grid topology, and the experimental results, here omitted for space constraints, show that CBP largely outperforms the other algorithms in all the considered scenarios, in terms of both percentage of successful matching and mean displacement error. Furthermore, the dependence of CBP performance on the node density and the shadowing variance is more predictable than for the other algorithms. We finally observe that, although in this study we focused on centralized algorithms, CBP can actually be realized in a distributed manner, though at non negligible communication costs. The investigation of this solution is one of the possible future developments of this work.

References

[1] Y. S. I. Akyildiz, W. Su and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks (Elsevier) Journal*, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[2] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: a quantitative comparison," *Computer Networks*, vol. 43, no. 4, pp. 499–518, 2003.

[3] N. Patwari, A. Hero III, M. Perkins, N. Correal, and R. O'dea, "Relative location estimation in wireless sensor networks," *Signal Processing, IEEE Transactions on*, vol. 51, no. 8, pp. 2137–2148, 2003.

[4] R. P. M. Eiman Elnahrawy, Xiaoyan Li, "The limits of localization using signal strength: a comparative study," in *Proceeding of Sensor and Ad Hoc Communications and Networks, IEEE SECON*, October 2004, pp. 406–414.

[5] G. Zanca, F. Zorzi, A. Zanella, and M. Zorzi, "Experimental comparison of RSSI-based localization algorithms for indoor wireless sensor networks," in *Proceedings of the workshop on Real-world Wireless Sensor Networks*. ACM, 2008, pp. 1–5.

[6] N. Patwari, J. Ash, S. Kyperountas, I. Hero, A.O., R. Moses, and N. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 54–69, July 2005.

[7] A. Savvides, M. Srivastava, L. Girod, and D. Estrin, "Localization in sensor networks," pp. 327–349, 2004.

[8] R. Huang and G. V. Zaruba, "Static path planning for mobile beacons to localize sensor networks," *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference on*, pp. 323–330, March 2007.

[9] J. Costa, N. Patwari, and A. Hero III, "Achieving high-accuracy distributed localization in sensor networks," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, vol. 3. IEEE, pp. iii–641.

[10] H. Wymeersch, J. Lien, and M. Win, "Cooperative localization in wireless networks," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 427 –450, feb. 2009.

[11] A. Ihler, J. Fisher III, R. Moses, and A. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *Selected Areas in Communications, IEEE Journal on*, vol. 23, no. 4, pp. 809–819, 2005.

[12] A. Goldsmith, *Wireless Communications*. New York, NY, USA: Cambridge University Press, 2005.

[13] A. Bardella, N. Bui, A. Zanella, and M. Zorzi, "An Experimental Study on IEEE 802.15. 4 Multichannel Transmission to Improve RSSI–Based Service Performance," *Real-World Wireless Sensor Networks*, pp. 154–161, 2010.

[14] R. Sinkhorn and P. Knopp, "Concerning nonnegative matrices and doubly stochastic matrices," *Pacific J. Math*, vol. 21, no. 2, pp. 343–348, 1967.

[15] F. Kschischang, B. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *Information Theory, IEEE Transactions on*, vol. 47, no. 2, pp. 498–519, 2001.

[16] L. Hans-Andrea, "An introduction to factor graphs," *IEEE Signal Processing Magazine*, vol. 21, pp. 28–41, 2004.

[17] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *Proceedings of Uncertainty in AI*, 1999, pp. 467–475.

[18] T. Richard, E. and S. Maneesh, "Demodulation as probabilistic inference," *Audio, Speech and Language Processing, IEEE Transactions on*, vol. 20, no. 10, 2011.

[19] W. Yair and F. William, T., "On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs," 2001.

[20] S. Vladimir and S. Zazo, "Sensor localization using nonparametric generalized belief propagation in network with loops," *Information Fusion, 2009. FUSION '09. 12th International Conference on.*, pp. 1966–1973, 2009.

[21] B. Kalantari, I. Lari, F. Ricca, and B. Simeone, "On the complexity of general matrix scaling and entropy minimization via the RAS algorithm", volume = 112, year = 2008, *Mathematical Programming*, pp. 371–401.