Distributed Sensor Location through Linear Programming with Triangle Inequality Constraints

Camillo Gentile
National Institute of Standards and Technology
Wireless Communication Technologies Group
Email: camillo.gentile@nist.gov

***Abstract*— Interest in dense sensor networks due to falling price and reduced size has motivated research in sensor location in recent years. To our knowledge, the algorithm which achieves the best performance in sensor location solves an optimization program by relaxing the quadratic geometrical constraints of the network to render the program convex. In recent work we proposed solving the same program, however by applying convex geometrical constraints directly, necessitating no relaxation of the constraints and in turn ensuring a tighter solution. This paper proposes a distributed version of our algorithm which achieves the same globally optimal objective function as the decentralized version. We conduct extensive experimentation to substantiate the robustness of our algorithm even in the presence of high levels of noise, and report the messaging overhead for convergence.**

***Index Terms*— Simplex method, Primal-dual method**

I. Introduction

The falling price and reduced size of sensors in recent years have fueled the deployability of dense networks to monitor and relay environmental properties such as temperature, moisture, and light [1]. The ability to self-organize and find their locations autonomously and with high accuracy proves particularly useful in military and public safety operations. In dense networks, multilateration can render good location accuracy despite significant errors in range estimates between sensors. This has launched a research area known as sensor location which seeks to process potentially enormous quantities of data collectively to achieve optimal results. Most practical systems require local distributed processing to cope with dynamic links or nodes in motion to maintain a network updated; alternatively relaying information across a large network sanctions the centralized processing of obsolete data, limiting scalability.

A recent paper on sensor location [2] provides an exhaustive survey of the available techniques for sensor location [3], [4], [6], [5]. To our knowledge, the two algorithms achieving the best performance in sensor location formulate a program with quadratic constraints to minimize a linear objective function [2], [7]. Since some of the constraints are non-convex, the papers differ primarily in their relaxation approaches to render the problem convex. The solution provided by Biswas et al. has greater applicability and yields better results than the one by Doherty et al. In recent work [8], we formulated a novel problem following their same approach, maintaining the efficiency of convex optimization, however by applying *linear* triangle inequality constraints as opposed to quadratic ones. This renders the problem automatically convex, necessitating no relaxation of the constraints and so guaranteeing a tighter solution, as confirmed through simulation.

This paper proposes a distributed algorithm for sensor location which converges to the optimal solution to the problem defined in [8]. Section II states this problem formally. The *primal-dual method* explained in Section III claims a key advantage over the conventional *simplex method* to solving the problem in a decentralized fashion. Exploiting this advantage leads to our distributed version of the primal-dual method described in Section IV. We demonstrate the steps of this distributed algorithm through a simple example network in Section V. An extensive number of challenging tests conditions are reported in Section VI to substantiate the robustness of our algorithm to high levels of noise in comparison to the algorithm proposed by Biswas. We also report the messaging overhead of the algorithm. The last section provides conclusions and directions for further research.

II. Preliminaries

Consider a network with two types of nodes: n_A anchor nodes (or anchors) with known location and n_S sensor nodes (or sensors) with unknown location, for a total of $n = n_A + n_S$ nodes. For simplicity, let the nodes lie on a plane such that node i has location $\mathbf{x}_i \in \mathcal{R}^2$ indexed through $i,\ i = 1 \ldots n_A$ for the anchors and $i = n_A + 1 \ldots n$ for the sensors. The set N contains all pairs of nodes between which a link exists: $(i, j),\ i < j;\ ||\mathbf{x}_i - \mathbf{x}_j|| < R$, where $||\cdot||$ denotes the Euclidean distance and the network parameter R is known as the *radio range*. The set M contains all triplets of nodes which form a triangle in the network: $(i, j, k),\ (i, j) \in N;\ (j, k) \in N;\ (i, k) \in N$.

Neighboring nodes i and j measure the link distance $\hat{d}_{ij}$ between them through received-signal-strength or time-of-arrival techniques [9]. Given the locations of the anchor nodes and the measured distances between neighboring nodes in the network, this paper considers the following problem to solve for the locations of the sensors:

$$\min \sum_{(i,j)\in N} |\alpha_{ij}|$$
s.t.
$$\left.\begin{array}{l} d_{ij}+d_{jk}\geq d_{ik} \\ d_{ij}+d_{ik}\geq d_{jk} \\ d_{jk}+d_{ik}\geq d_{ij}\end{array}\right\},\ \forall(i,j,k)\in M \tag{1}$$

where $d_{ij}=\hat{d}_{ij}+\alpha_{ij}$.

The problem minimizes the sum of the absolute *residuals* α_{ij} between the measured distances $\hat{d}_{ij}$ and the estimated distances d_{ij} such that the latter conform to requisite geometrical constraints. Using triangle inequality constraints as opposed to quadratic constraints [2], [7] ensures the convexity of the problem without relaxing any of the original geometrical constraints. Rewriting the problem in standard form removes the absolute values and introduces bounding constraints:

$$\min \sum_{(i,j)\in N} \alpha^+_{ij}+\alpha^-_{ij}$$
s.t.
$$\left.\begin{array}{l} d_{ij}+d_{jk}\geq d_{ik} \\ d_{ij}+d_{ik}\geq d_{jk} \\ d_{jk}+d_{ik}\geq d_{ij}\end{array}\right\},\ \forall(i,j,k)\in M \tag{2}$$
$$\left.\begin{array}{l} \alpha^+_{ij}\geq 0 \\ \alpha^-_{ij}\geq 0\end{array}\right\},\ \forall(i,j)\in N$$

where $\alpha_{ij}=\alpha^+_{ij}-\alpha^-_{ij}$. The solution to the problem above does not directly yield the sensor locations $\mathbf{x}_i,\ i=n_A+1...n$, but only the values of the residuals of the link distances. Hence the complete algorithm requires aposteriori *location propagation* described in [8] to furnish the locations of the sensors from the residuals.

Theorem: If any one of the three inequality constraints of a triangle is bound, then the other two are feasible.

Proof: Assume without loss of generality that the first inequality constraint is bound:

$$d_{ij}+d_{jk}=d_{ik}\Rightarrow\left\{\begin{array}{l} -d_{ij}+d_{ik}=d_{jk} \\ -d_{jk}+d_{ik}=d_{ij}\end{array}\right.,$$
$$\text{but}\left\{\begin{array}{l} d_{ij}+d_{ik}\geq -d_{ij}+d_{ik} \\ d_{jk}+d_{ik}\geq -d_{jk}+d_{ik}\end{array}\right.\quad \text{since } d_{ij},d_{jk}\geq 0,$$
$$\text{so}\left\{\begin{array}{l} d_{ij}+d_{ik}\geq d_{jk} \\ d_{jk}+d_{ik}\geq d_{ij}\end{array}\right..\ \square \tag{3}$$

III. The primal-dual method

A. The primal problem

We denote the linear program (2) as the *primal* problem. Rewriting the primal in canonical form appears as

$$\min \sum_{(i,j)\in N} \alpha^+_{ij}+\alpha^-_{ij}$$
s.t.
$$\left.\begin{array}{l} a^1_{ij,k}(\alpha^+_{ij}\text{-}\alpha^-_{ij})+a^1_{jk,i}(\alpha^+_{jk}\text{-}\alpha^-_{jk})+a^1_{ik,j}(\alpha^+_{ik}\text{-}\alpha^-_{ik})\text{-}\alpha^1_{ijk}=b^1_{ijk} \\ a^2_{ij,k}(\alpha^+_{ij}\text{-}\alpha^-_{ij})+a^2_{jk,i}(\alpha^+_{jk}\text{-}\alpha^-_{jk})+a^2_{ik,j}(\alpha^+_{ik}\text{-}\alpha^-_{ik})\text{-}\alpha^2_{ijk}=b^2_{ijk} \\ a^3_{ij,k}(\alpha^+_{ij}\text{-}\alpha^-_{ij})+a^3_{jk,i}(\alpha^+_{jk}\text{-}\alpha^-_{jk})+a^3_{ik,j}(\alpha^+_{ik}\text{-}\alpha^-_{ik})\text{-}\alpha^3_{ijk}=b^3_{ijk}\end{array}\right\},\ \forall(i,j,k)\in M$$
$$\left.\begin{array}{l} \alpha^+_{ij}\geq 0 \\ \alpha^-_{ij}\geq 0\end{array}\right\},\ \forall(i,j)\in N \tag{4}$$

where

$$\begin{bmatrix} a^1_{ij,k} & a^1_{jk,i} & a^1_{ik,j} \\ a^2_{ij,k} & a^2_{jk,i} & a^2_{ik,j} \\ a^3_{ij,k} & a^3_{jk,i} & a^3_{ik,j}\end{bmatrix}=\begin{bmatrix} 1 & 1 & \text{-}1 \\ 1 & \text{-}1 & 1 \\ \text{-}1 & 1 & 1\end{bmatrix},\quad \begin{bmatrix} b^1_{ijk} \\ b^2_{ijk} \\ b^3_{ijk}\end{bmatrix}=\begin{bmatrix} \text{-}1 & \text{-}1 & 1 \\ \text{-}1 & 1 & \text{-}1 \\ 1 & \text{-}1 & \text{-}1\end{bmatrix}\begin{bmatrix} \hat{d}_{ij} \\ \hat{d}_{jk} \\ \hat{d}_{ik}\end{bmatrix},$$

$\alpha^+_{ij},\alpha^-_{ij}$ are the $2|N|$ primal *decision* variables, and $\alpha^1_{ijk},\alpha^2_{ijk},\alpha^3_{ijk}$ are the $3|M|$ primal *slack* variables.

A *basic* solution to the primal problem contains exactly $3|M|$ nonzero (basic) variables and $2|N|$ zero (nonbasic) variables for the nongenerate case which assumes linear independence of all constraints in (4). Since the optimal solution is basic [10], the *simplex method* pivots between basic feasible solutions of the system to find it. A *pivot* consists of raising an *entering* variable in the nonbasic set from zero that will improve the objective function. The entering variable can rise only a certain amount until a *blocking* variable in the basic set reduces to zero; hence the entering variable becomes basic and the blocking variable nonbasic at another basic feasible solution of the system. Determining this amount necessitates global knowledge of the values of all the basic variables, such that no variable loses feasibility and exactly $2|N|$ of them equal zero throughout the pivots. Hence the simplex method does not lend to distributed processing. *Interior-point methods* are suited for centralized computing even more than the simplex method, requiring the inversion of large sparse symmetric matrices [11].

B. The dual problem

Each primal linear program has a unique *dual* linear program. The dual problem to (4) appears as [12]:

$$\max \sum_{(i,j,k)\in M} b^1_{ijk}\beta^1_{ijk}+b^2_{ijk}\beta^2_{ijk}+b^3_{ijk}\beta^3_{ijk}$$
s.t.
$$\left.\begin{array}{ll} \alpha^+_{ij}: & \sum_{k,\ (i,j,k)\in M} a^1_{ij,k}\beta^1_{ijk}+a^2_{ij,k}\beta^2_{ijk}+a^3_{ij,k}\beta^3_{ijk}+\beta^+_{ij}=1 \\ \alpha^-_{ij}: & \sum_{k,\ (i,j,k)\in M} \text{-}a^1_{ij,k}\beta^1_{ijk}\text{-}a^2_{ij,k}\beta^2_{ijk}\text{-}a^3_{ij,k}\beta^3_{ijk}+\beta^-_{ij}=1\end{array}\right\},\ \forall(i,j)\in N$$
$$\left.\begin{array}{ll} \alpha^1_{ijk}: & \beta^1_{ijk}\geq 0 \\ \alpha^2_{ijk}: & \beta^2_{ijk}\geq 0 \\ \alpha^3_{ijk}: & \beta^3_{ijk}\geq 0\end{array}\right\},\ \forall(i,j,k)\in M \tag{5}$$

where $\beta^1_{ijk}, \beta^2_{ijk}, \beta^3_{ijk}$ are the dual decision variables and $\beta^+_{ij}, \beta^-_{ij}$ are the dual slack variables. As indicated, each primal variable has a corresponding dual constraint.

The *Complementary Slackness Theorem* [10] states that *any* feasible primal and dual solutions are optimal if the following complementary slackness conditions hold:

$$\begin{array}{lll} (a) & \alpha^u_{ijk} = 0 \;\Rightarrow\; \beta^u_{ijk} \geq 0 & \multirow{2}{*}{} \\ (b) & \alpha^u_{ijk} > 0 \;\Rightarrow\; \beta^u_{ijk} = 0 & \end{array}\Bigg\},\ \Bigg\{ \begin{array}{l} \forall (i,j,k) \in M \\ \forall u \in \{1,2,3\} \end{array}$$
$$\begin{array}{lll} (c) & \alpha^v_{ij} = 0 \;\Rightarrow\; \beta^v_{ij} \geq 0 \\ (d) & \alpha^v_{ij} > 0 \;\Rightarrow\; \beta^v_{ij} = 0 \end{array}\Bigg\},\ \Bigg\{ \begin{array}{l} \forall (i,j) \in N \\ \forall v \in \{+,\text{-}\} \end{array} \tag{6}$$

Rather than solve the primal problem through the simplex method, we formulate a distributed version of the *primal-dual method*. The key advantage of the latter relaxes the condition that a primal solution be basic. Our algorithm proceeds in the follow manner: first a link in the network finds a *local* feasible primal solution (not necessarily basic) such that all incident triangles meet the triangle inequality constraints. Then the link applies the complementary slackness conditions given through this primal solution, defining the local *restricted* dual problem. If the dual solution to the restricted problem is also feasible, then the local primal solution is optimal through the Complementary Slackness Theorem; otherwise the primal solution is modified. Once all the links attain compatible locally compatible optimal solutions, the network achieves the globally optimal solution. Dantzig treats a full discussion on the primal-dual method [11].

IV. Distributed location

A. Network organization

The nodes in the network transmit asynchronously. If node i wakes up after node j, then n_i manages the link ℓ_{ij}. The link manager maintains the information on ℓ_{ij}: the measured distance $\hat{d}_{ij}$ and the residual α_{ij} initialized to zero. Fig. 1a illustrates an example network with three nodes n_i, n_j, n_k, where n_k woke up first and n_j second, assigning to n_j manager of ℓ_{jk}. When n_i wakes up subsequently, it broadcasts a HELLO message containing its ID#: i. In Fig. 1b, nodes n_j and n_k respond with their ID#s; since n_j serves as link manager, it also broadcasts the information on the links it manages. In receiving the messages from n_j and n_k in Fig. 1c, n_i becomes manager of ℓ_{ij} and ℓ_{ik} and estimates $\hat{d}_{ij}$ and $\hat{d}_{ik}$; the two-way message exchange allows n_i to measure these distances asynchronously [13]. As manager, n_i then broadcasts the information on the links it manages. Now both managers in Fig. 1d have access to information on all three links of $\triangle_{ijk}$. While we refer to the links as the processing centers for the distributed algorithm in the sequel, the actual processing of course takes place at the link managers.

B. A feasible primal solution

Denote a triangle $\triangle_{ijk}$ as *feasible* if it meets all three of its triangle inequality constraints: $\alpha^u_{ijk} \geq 0,\ \forall u \in \{1,2,3\}$.

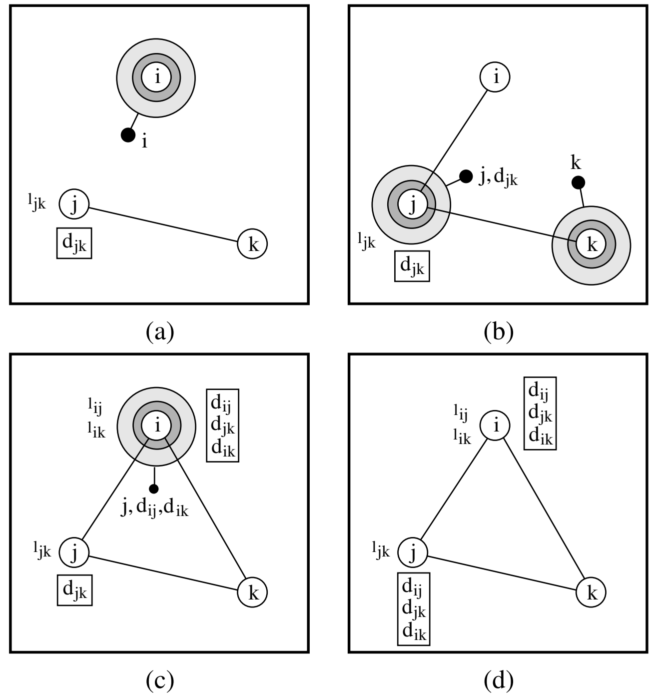


Fig. 1. Transmitted messages in network organization.

Suppose that a link ℓ_{ij} changes value, rendering one of its incident triangles infeasible: $\alpha^u_{ijk} < 0$, for u, k. Proof (3) shows that if any one of the constraints is bound, then the other two are feasible; so by setting $\alpha^u_{ijk} = 0$, ℓ_{ij} restores feasibility to $\triangle_{ijk}$. Since the value of ℓ_{ij} was just changed, it remains the same; rather the link selects one of the other two links on the triangle (ℓ_{jk} or ℓ_{ik}) to set the value of α^u_{ijk} to zero, say ℓ_{jk}. Consider modifying α^v_{jk} such that $\alpha'_{jk} = \alpha_{jk} + \left(\frac{\partial \alpha^v_{jk}}{\partial \alpha^u_{ijk}}\right)\delta\alpha^u_{ijk}$, where $\delta\alpha^u_{ijk} = -\alpha^u_{ijk}$ represents the necessary change in α^u_{ijk} to render the violated constraint feasible. The partial

$$\left(\frac{\partial \alpha^v_{jk}}{\partial \alpha^u_{ijk}}\right)_{\alpha_{ij},\alpha_{ik}} = \frac{1}{v a^u_{jk,i}} \tag{7}$$

while maintaining the value of the links ℓ_{ij} and ℓ_{ik} constant is computed by rewriting the primal constraint u in (4) as

$$a^u_{ij,k}\underbrace{(\alpha^+_{ij}\text{-}\alpha^-_{ij})}_{\alpha_{ij}} + a^u_{jk,i}\overbrace{(\alpha^+_{jk}\text{-}\alpha^-_{jk})}^{v\alpha^v_{jk}} + a^u_{ik,j}\underbrace{(\alpha^+_{ik}\text{-}\alpha^-_{ik})}_{\alpha_{ik}} \text{-}\,\alpha^u_{ijk} = b^u_{ijk}. \tag{8}$$

Note that $\alpha^v_{jk} > 0$ implies $\alpha^{\text{-}v}_{jk} = 0$. The program below summarizes the local pivot:

$$
\boxed{
\begin{array}{l}
\text{Pivot I: Set } \alpha^u_{ijk} = 0 \text{ by modifying } \alpha^v_{jk} \\[6pt]
\alpha'_{jk} = \alpha_{jk} + \left(\frac{\partial \alpha^v_{jk}}{\partial \alpha^u_{ijk}}\right) \delta \alpha^u_{ijk} \\
\left(\frac{\partial \alpha^v_{jk}}{\partial \alpha^u_{ijk}}\right)_{\alpha_{ij}, \alpha_{ik}} = \frac{1}{va^u_{jk,i}} \\
\delta \alpha^u_{ijk} = -\alpha^u_{ijk}
\end{array}
}
\tag{9}
$$

In restoring feasibility to $\triangle_{ijk}$, ℓ_{jk} may in turn render a neighboring triangle $\triangle_{jkl}$ infeasible, analagous to the change in ℓ_{ij} which rendered $\triangle_{ijk}$ infeasible in the previous step. Through this mechanism infeasibility *propagates* through the network between triangles, leaving those in its path feasible, and so obtaining a feasible primal solution locally at each step until termination at a certain triangle. In the worst case, propagation terminates at the edge of the network where ℓ_{jk} has no neighboring $\triangle_{jkl}$. Section $IV - C$ describes how to select ℓ_{jk} optimally.

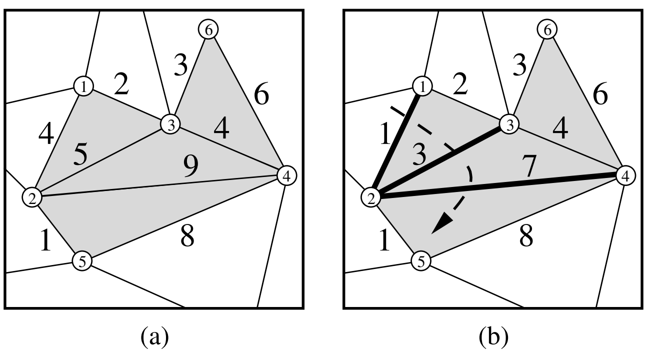


Fig. 2. Propagation in finding a feasible primal solution.

The portion of a network in Fig. 2a consists of four feasible triangles (shaded) with the estimated distances $d = \hat{d}$ displayed on each link. Let d_{12} decrease to 1, rendering $\triangle_{123}$ infeasible. The infeasibility then propagates along the path indicated by the dashed arrow in Fig. 2b: d_{23} decreases to 3, restoring feasibility to $\triangle_{123}$, but rendering $\triangle_{234}$ infeasible; d_{24} decreases to 7, restoring feasibility to $\triangle_{234}$ while maintaining $\triangle_{245}$ feasible. The propagation terminates at $\triangle_{245}$ with all four triangles newly feasible.

C. The restricted dual problem

1) Defining and solving the restricted dual problem: We say that a link ℓ_{ij} has a local feasible primal solution if all its incident triangles $\triangle_{ijk}$, $\forall k$, $(i, j, k) \in M$ are feasible. Once a link obtains a local feasible primal solution, it applies the complementary slackness conditions which define the local restricted dual: every bound primal constraint $\alpha^u_{ijk} = 0$ admits one dual decision variable (unknown) $\beta^u_{ijk} \geq 0$ to the restricted problem (6a), while setting the remaining dual decision variables to zero (6b); every nonzero primal decision variable $\alpha^v_{ij} > 0$ admits one bound dual constraint (equation) $\beta^v_{ij} = 0$ (6d), while unbounding the remaining dual constraints (6c). As any solution to a nondegenerate linear system contains at least the same number of unknowns as equations, any solution to the nondegenerate primal system in (4) contains at least the same number of nonzero primal decision variables as the number of bound primal constraints. So due to complementary slackness, the restricted dual contains no more unknowns than equations, allowing to solve for the unknowns through simple substitution.

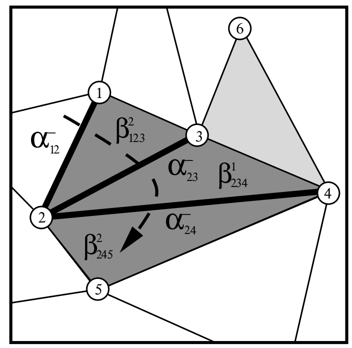


Fig. 3. Propagation in solving the restricted dual problem.

Fig. 3 graphically represents the restricted dual problem corresponding to the feasible primal solution in Fig. 2b. Each of the three darkly shaded triangles has one bound primal constraint, admitting one unknown per triangle to the dual: β^2_{123} ($d_{12} + d_{13} = d_{23}$) at $\triangle_{123}$, β^1_{234} ($d_{23} + d_{34} = d_{24}$) at $\triangle_{234}$, and β^2_{245} ($d_{24} + d_{25} = d_{45}$) at $\triangle_{245}$; each of the three boldfaced links with nonzero residuals ($\alpha^-_{12} = 3, \alpha^-_{23} = 2, \alpha^-_{24} = 2$) admits one equation per link to the dual from (5): ($-\beta^2_{123} = 1$) at ℓ_{12}, ($\beta^2_{123} - \beta^1_{234} = 1$) at ℓ_{23}, and ($\beta^1_{234} - \beta^2_{245} = 1$) at ℓ_{24}. Although ℓ_{23} and ℓ_{24} cannot solve for their two unknowns, ℓ_{12} can solve for its single unknown $\beta^1_{123} = -1$, and then propagates the value to the other two links of $\triangle_{123}$; now knowing the value of β^2_{123}, l_{23} solves for its single remaining unknown $\beta^1_{234} = 2$ and propagates the value to the other two links of $\triangle_{234}$; now knowing the value of β^1_{234}, ℓ_{24} solves for its single remaining unknown $\beta^2_{245} = 1$ and propagates the value to the other two links of $\triangle_{245}$. Paralleling the primal solution, the dual solution is also found through propagation. The dashed arrow in Fig. 3 indicates the direction of propagation.

2) Modifying the primal solution towards optimality: If the solution to the restricted dual is feasible (i.e. all the decision and slack variables are greater than or equal to zero), then the primal solution is optimal. Otherwise it can be shown [11] that setting an infeasible dual variable to zero improves the primal objective, provided that the nonzero feasible dual variables remain admitted to the restricted dual problem.

If the restricted dual solution includes an infeasible decision variable $\beta^u_{ijk} < 0$, then raising its corresponding primal slack variable α^u_{ijk} from zero sets $\beta^u_{ijk} = 0$ through complementary slackness. However entering α^u_{ijk} lowers a local blocking variable $\alpha^{\tilde{u}}_{ijk} > 0, \tilde{u} \neq u, \frac{\partial \alpha^u_{ijk}}{\partial \alpha^{\tilde{u}}_{ijk}} = -1$; conversely, reducing $\alpha^{\tilde{u}}_{ijk}$ to zero through Pivot I (9) raises α^u_{ijk}. The partial

$$\left(\frac{\partial \alpha^u_{ijk}}{\partial \alpha^{\tilde{u}}_{ijk}}\right)_{\alpha_{ij},\alpha_{ik}} = \left(\frac{\partial \alpha^v_{jk}}{\partial \alpha^{\tilde{u}}_{ijk}}\right)\left(\frac{\partial \alpha^u_{ijk}}{\partial \alpha^v_{jk}}\right) = \frac{a^u_{jk,i}}{a^{\tilde{u}}_{jk,i}} \tag{10}$$

is computed through (7). Note that Pivot I results in $\alpha^v_{jk} > 0$, and in turn sets $\beta^v_{jk} = 0$ through complementary slackness; so if $\beta^v_{jk} > 0$ before the pivot, the pivot removes a nonzero feasible dual variable from the restricted dual. Select α^v_{jk} such that $\beta^v_{jk} \leq 0$.

If the restricted dual solution includes an infeasible slack variable $\beta^v_{ij} < 0$, then raising its corresponding primal decision variable α^v_{ij} from zero sets $\beta^v_{ij} = 0$ through complementary slackness. However entering α^v_{ij} lowers a local blocking variable $\alpha^{\tilde{v}}_{jk} > 0, \frac{\partial \alpha^v_{ij}}{\partial \alpha^{\tilde{v}}_{ij}} = -1$; conversely, reducing $\alpha^{\tilde{v}}_{jk}$ to zero through Pivot II below raises α^v_{ij}. The partial $\frac{\partial \alpha^v_{ij}}{\partial \alpha^{\tilde{v}}_{jk}}$ is computed through (8).

Pivot II: Set $\alpha^{\tilde{v}}_{jk} = 0$ by raising α^v_{ij}

$$\begin{aligned} &\alpha'_{ij} = \alpha_{ij} + \left(\frac{\partial \alpha^v_{ij}}{\partial \alpha^{\tilde{v}}_{jk}}\right)\delta\alpha^{\tilde{v}}_{jk}, \\ &\left(\frac{\partial \alpha^v_{ij}}{\partial \alpha^{\tilde{v}}_{jk}}\right)_{\alpha_{ik},\alpha^u_{ijk}=0} = -\frac{\tilde{v} a^u_{jk,i}}{v a^u_{ij,k}}, \\ &\delta\alpha^{\tilde{v}}_{jk} = -\alpha^{\tilde{v}}_{jk} \end{aligned} \tag{11}$$

Note that Pivot II affects the value of $\alpha^u_{ijk}, \forall u \in \{1, 2, 3\}$; so if $\beta^u_{ijk} > 0$ before the pivot, maintain $\alpha^u_{ijk} = 0$ such that the nonzero feasible dual variable remains in the restricted problem.

Recall in Section IV-B that the initial primal feasible solution is found in the absence of any dual restrictions, and so the arbitrary selection of the link to modify in Pivot I may lead to a sub-optimal solution. When modifying the primal solution towards optimality, the local pivot may affect the primal feasibility of a non-incident triangle on the link. And so another link on that triangle maneuvers to restore its feasibility through Pivot I, but now using the local dual restrictions to decide which one. The new primal feasible solution of that triangle in turn generates other dual restrictions local to the selected link. As the pivots continue, the global primal problem becomes more and more restricted by the local duals until achieving global optimality. An example provided in the following section substantiates these ideas.

D. Location propagation

The estimated distances computed through (4) yield the desired sensor locations through *location propagation* [14]. The anchor nodes propagate their locations to the sensor nodes in a distributed fashion: if two anchor nodes share a neighboring sensor node, the sensor location can be determined from the two anchor locations coupled with the two estimated distances between the anchors and the sensor. Once the sensor location is *known*, it serves with another known sensor (or anchor) to determine the location of an unknown sensor neighboring the two. Fig. 4 displays two anchor nodes (shaded) and four sensor nodes. Anchors n_1 and n_2 propagate their locations to unknown sensor n_3; anchor n_2 and now known sensor n_3 propagate their locations to unknown sensor n_4, anchor n_2 and now known sensor n_4 propagate their locations to unknown sensor n_5, and known sensor n_3 and known sensor n_4 propagate their locations to unknown sensor n_6. The location propagation is embedded with the primal-dual pivots in our distributed location algorithm. The details as described in [8] and are omitted for space reasons.

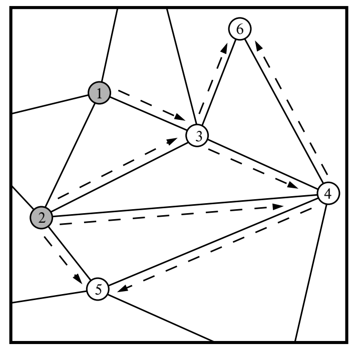
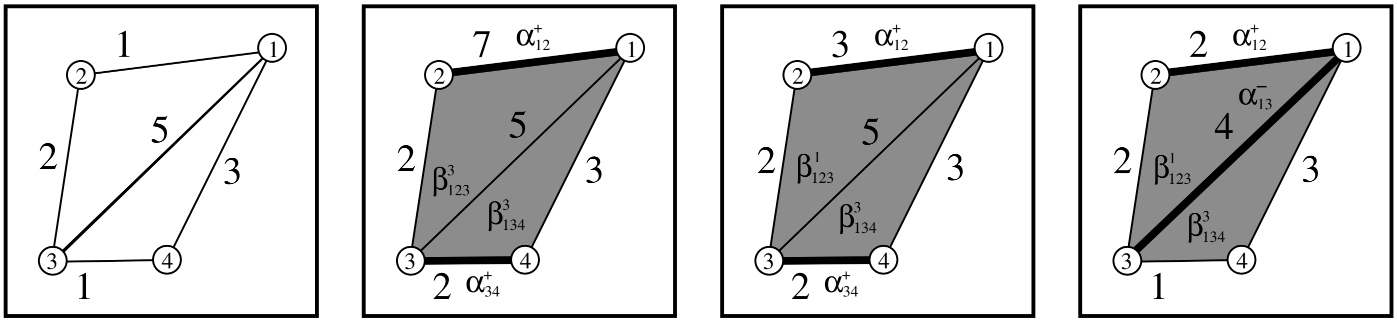


Fig. 4. Propagation in determining the sensor locations.

V. Example network

Consider the example network in Fig. 0 of Table I with four nodes and five links. The measured distances $\hat{d}$ appear on each link. Table I displays the coefficients of the primal problem corresponding to this example network in $\mathbf{A}$ and $\mathbf{b}$. The blank slots indicate zeros and are left so to reduce clutter. As the nodes wake up asynchronously, assume that links ℓ_{12} and ℓ_{34} are the last two established, completing the triangles $\triangle_{123}$ and $\triangle_{234}$ respectively. Note that $\triangle_{123}$ and $\triangle_{234}$ are initially infeasible with $\alpha^1_{123} = -2$ and $\alpha^3_{134} = -1$. In solving the initial primal problem and in the absence of any dual restrictions, ℓ_{12} raises its value to $7(\alpha^+_{12} = 6)$ to restore feasibility to $\triangle_{123}$, and distributes this value to the other two links; likewise ℓ_{34} raises its value to $2(\alpha^+_{34} = 1)$ to restore feasibility to $\triangle_{234}$, and distributes this value to the other two links. Now that each link ℓ_{ij} in the network holds the primal decision variables of links ℓ_{jk} and ℓ_{ik} for all incident triangles $\triangle_{ijk}$, it can compute the primal slack variables $\alpha^1_{ijk}, \alpha^2_{ijk}, \alpha^3_{ijk}$. The initial (1) primal solution with objective 7 appears in row $\alpha^v_{ij}(1)$ and column $\alpha^u_{ijk}(1)$ of Table I. The solution is indexed according to global solutions for the sake of clarity, but as just explained, the solutions are computed locally, distributedly, and asynchronously.

With all the triangles incident on ℓ_{12} now feasible, ℓ_{12} applies the complementary slackness conditions from its local primal solution to define the corresponding local restricted dual problem:



(0) (1) (2) (3)

A	α^+_{12}	α^-_{12}	α^+_{23}	α^-_{23}	α^+_{13}	α^-_{13}	α^+_{34}	α^-_{34}	α^+_{14}	α^-_{14}	**b**	$\alpha^u_{ijk}(1)$	$\beta^u_{ijk}(1)$	$\alpha^u_{ijk}(2)$	$\beta^u_{ijk}(2)$	$\alpha^u_{ijk}(3)$	$\beta^u_{ijk}(3)$
β^1_{123}	1	−1	1	−1	−1	1					2	4			$\boxed{1}$		$\boxed{1}$
β^2_{123}	1	−1	−1	1	1	−1					−4	10		4		4	
β^3_{123}	−1	1	1	−1	1	−1					−6		$\boxed{-1}$	4		4	
β^1_{234}					1	−1	1	−1	−1	1	−7	8		8		6	
β^2_{234}					1	−1	−1	1	1	−1	−3	2		2		2	
β^3_{234}					−1	1	1	−1	1	−1	1		$\boxed{1}$		$\boxed{1}$		$\boxed{0}$
$\alpha^v_{ij}(1)$	6			0			1		0								
$\beta^v_{ij}(1)$	$\Box$	2	2		3	−1	$\Box$	2		2							
$\alpha^v_{ij}(2)$	2		0				1		0								
$\beta^v_{ij}(2)$	$\Box$	2		2	3	−1	$\Box$	2		2							
$\alpha^v_{ij}(3)$	1		0			1											
$\beta^v_{ij}(3)$	$\Box$	2		2	2	$\Box$	1	1	1	1							

TABLE I
THE PRIMAL-DUAL TABLE

$$
\begin{aligned}
\alpha^3_{123} = 0 \;&\Rightarrow\; \beta^3_{123} \geq 0 \\
\{\alpha^1_{123}, \alpha^2_{123}\} > 0 \;&\Rightarrow\; \{\beta^1_{123}, \beta^2_{123}\} = 0 \\
\{\alpha^-_{12}, \alpha^+_{23}, \alpha^-_{23}, \alpha^+_{13}, \alpha^-_{13}\} = 0 \;&\Rightarrow\; \{\beta^-_{12}, \beta^+_{23}, \beta^-_{23}, \beta^+_{13}, \beta^-_{13}\} \geq 0 \\
a^+_{12} > 0 \;&\Rightarrow\; \beta^+_{12} = 0
\end{aligned}
$$

So ℓ_{12} processes a single unknown ($\beta^3_{123} \geq 0$) and a single equation ($-\beta^3_{123} = 1$) to solve for its local restricted dual. It solves for $\beta^3_{123} = -1$ and distributes this value to the other two links of $\triangle_{123}$. Through the same process, ℓ_{34} solves for $\beta^3_{234} = 1$ in its restricted dual and distributes this value to the other two links of the triangle. Now that each link ℓ_{ij} in the network holds the dual decision variables $\beta^1_{ijk}, \beta^2_{ijk}, \beta^3_{ijk}$ of all incident triangles $\triangle_{ijk}$, it can compute its slack dual variables $\beta^+_{ij}, \beta^-_{ij}$. The initial (1) dual solution appears in column $\beta^u_{ijk}(1)$ and row $\beta^v_{ij}(1)$ of Table I. The table evidences the complimentary slackness structure of the primal-dual solution, where the dual slack (decision) variable is zero if a primal decision (slack) variable is nonzero. The boxes indicate the admitted unknowns and equations in each restricted dual, and Fig. 1 graphically represents the restricted dual problem.

The initial dual solution reveals two infeasible variables $\beta^3_{123} = -1$ and $\beta^-_{13} = -1$. Raising α^3_{123} or α^-_{13} will improve the primal objective, provided that the nonzero feasible variables remain admitted to the restricted dual problem. Link ℓ_{12} raises α^3_{123} through Pivot I, setting the local blocking variable $\alpha^1_{123} = 4$, $\frac{\partial \alpha^3_{123}}{\partial \alpha^1_{123}} = -1$ to zero by modifying the value of ℓ_{12} to 3 ($\alpha^+_{12} = 2$). Selecting to remove $\beta^+_{12} = 0$ from the restricted dual problem ensures that this pivot improves the primal objective. The second primal solution remains feasible after Pivot I, necessitating no additional pivots to restore feasibility. Note that the primal objective equal to 3 has indeed improved. The second primal-dual solution appears in Table I. This dual solution still reveals the infeasible variable $\beta^-_{13} = -1$. Link ℓ_{13} raises α^-_{13} through Pivot II, setting the local blocking variable $a^+_{34} = 1$, $\frac{\partial \alpha^-_{13}}{\partial \alpha^+_{34}} = -1$ to zero while maintaining $\alpha^3_{134} = 0$ such that $\beta^3_{134} = 1$ remains admitted to the restricted dual problem. This local pivot however raises α^1_{123} from zero, violating a non-local dual restriction by removing $\beta^1_{123} = 1$ from the same restricted dual. Link ℓ_{13} sets α^1_{123} back to zero through Pivot I by modifying the value of ℓ_{12} to 2 ($\alpha^+_{12} = 1$). The third primal-dual solution with objective 2 appears in Table I, where all the feasible dual variables indicate the optimality of this primal solution.

VI. Experimental setup and results

In our previous work [8], we compared our centralized algorithm to Biswas by conducting experiments on a network with the same structure. The network contains 50 sensor nodes uniformly distributed throughout a one by one unit area. The three varying parameters are the number of anchor nodes, the radio range, and the noisy factor of the link distances. As Biswas, the ground-truth link distances $\bar{d}_{ij}$ between neighboring nodes i and j are perturbed with zero-mean unit-variance Gaussian noise $\mathcal{N}(0,1)$ and the varying parameter $noise$. So the link managers measure the noisy link distances $\hat{d}_{ij} = \bar{d}_{ij} * (1 + \mathcal{N}(0,1) * noise)$.

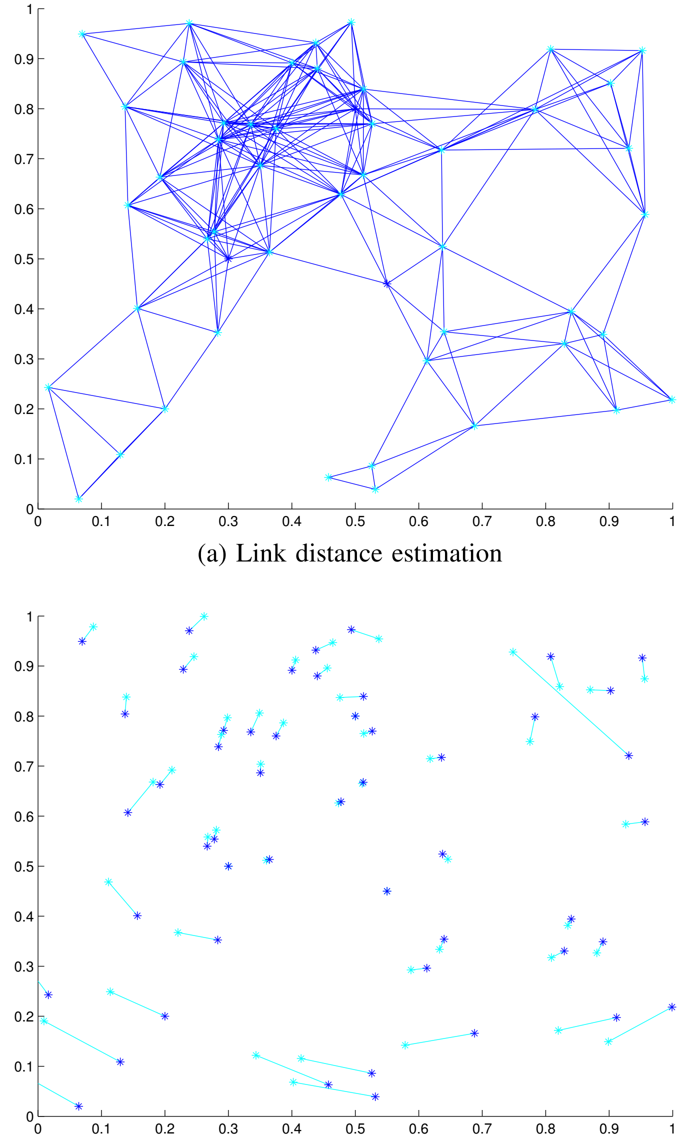
Fig. 5(a) illustrates a test network with three anchors, $R = 0.25$, and $noise = 0.1$. The anchors and sensors appear as dark and light asterisks respectively, and the links as dark lines between neighboring nodes. The network contains 225 links for an average node connectivity of 7.9623. The distributed location algorithm yields the estimated locations of the sensor nodes upon convergence. The true and estimated locations appear in Fig.5(b) as dark and light asterisks connected by an error line. The average location error is 0.0597.

Biswas reports the results of a single trial network for the six test conditions described in [2]. The quantative measure for each test condition is the average location error over the sensor nodes

$$\frac{1}{n_S} \sum_{i=n_A+1}^{n} ||\bar{\mathbf{x}}_i - \mathbf{x}_i||, \tag{12}$$

where $\bar{\mathbf{x}}_i$ and $\mathbf{x}_i$ denote the ground-truth and estimated locations. Our paper includes a more extensive superset of their test conditions, spanning a much higher range of noise, for a total of 38 tests. In addition, for each test we conduct ten trials of randomly distributed sensor networks rather than one, totaling 380 trials. The result for each test condition is reported as the average over the ten trials. Table II contains the results for 36 tests as the cross product of $\#anchor = \{3, 5, 7\}$, $R = \{0.20, 0.25, 0.30\}$, and $noise = \{0.0, 0.1, 0.2, 0.3\}$, and two additional tests $(7, 0.30, 0.05)$ and $(7, 0.40, 0.1)$. The average connectivity of the networks for three anchors is 5.4372 for $R = 0.20$, 7.7238 for $R = 0.25$, and 10.2477 for $R = 0.30$. For each slot in the table, the average location error of our distributed is reported on the top line, and if available, the corresponding result in [2] is shown in parentheses on the middle line in the same slot. Our algorithm delivers up to three times better performance across all parameters. In fact in the column for seven anchor nodes and $R = 0.30$, their error 0.0640 for $noise = 0.1$ is still 39% greater than our error 0.0459 for $noise = 0.3$; this shows that our algorithm is much more robust to noise.

Our results presented here match up exactly with the results from our centralized algorithm in [8], confirming that the distributed primal-dual algorithm indeed attains global optimality. The bottom line of each slot in Table II displays the average number of transmitted messages required for our distributed algorithm to converge, including network organization, primal-dual pivots, and location propagation. Observe the number of messages for network organization and location propagation in the row for $noise = 0.0$; here all the triangles are initially feasible, necessitating no primal-dual pivots. The primal-dual messages increases from zero with increasing noise due to the number of initially infeasible triangles, from a fraction more for $noise = 0.1$ up to twice more for very high levels of $noise = 0.3$ and $R = 0.3$. The messages also increases with increasing radio range R due to the increased number of triangles in the network, but also delivers better location performance. Introducing more anchors in the network causes a mild decrease in location propagation messages since more anchor nodes can reach the same number of sensors.



(a) Link distance estimation

(b) Location reconstruction

Fig. 5. A test network with three anchor nodes, R=0.25, and $noise = 0.1$.

VII. Conclusions and further work

This paper proposes a distributed version of an algorithm for sensor location described in our recent work. Drawing

noise	R=0.20			R=0.25			R=0.30			R=0.4
	3	5	7	3	5	7	3	5	7	7
0.0	0.0427 (0.0800) 377.2	0.0419 369.5	0.0408 366.0	0.0067 (0.0076) 433.4	0.0058 421.6	0.0058 412.3	$< 1e^{-6}$ $(1.8e^{-4})$ 484.1	$< 1e^{-6}$ 475.1	$< 1e^{-6}$ 462.8	
0.05									0.0162 (0.0540) 577.6	
0.1	0.0754 395.1	0.0644 388.4	0.0638 382.2	0.0526 372.4	0.0436 463.4	0.0270 457.2	0.0447 667.4	0.0362 657.3	0.0245 (0.0640) 643.1	0.0114 (0.0500) 1368.2
0.2	0.0846 422.9	0.0801 415.8	0.0676 409.7	0.0764 556.2	0.0649 539.3	0.0493 527.5	0.0570 1051.2	0.0566 1021.7	0.0458 1015.5	
0.3	0.1063 455.6	0.0877 453.9	0.0873 444.6	0.0954 847.0	0.0825 833.7	0.0772 827.9	0.0767 1405.4	0.0736 1384.3	0.0459 1373.1	

TABLE II

NUMERICAL RESULTS FOR EXPERIMENTS

on previous approaches employing complex optimization, our approach provides a tighter solution to the problem than its competitors by applying triangle inequality geometrical constraints to the network. In order to substantiate its performance, we run an extensive set of experiments in comparison with the published results for the best competing algorithm. We report the number of distributed messages for convergence of our algorithm, and show that it delivers the exact same results as the centralized version, and so too attains the optimal objective function.

Supplementary work not included in the paper demonstrates that our novel distributed algorithm reorganizes efficiently in networks when the measured distances change over time, or the original links break or new links are added.

REFERENCES

[1] P.F. Gorder, "Sizing up smart dust," *IEEE Journal on Computing in Sciences and Engineering*, vol. 5, no. 6, pp. 6-9, Nov. 2003.

[2] P. Biswas and Y. Yec "Semidefinite Programming for Ad Hoc Wireless Sensor Network Localization," *IEEE Conf. on Information Processing in Sensor Networks* pp. 46-54, April 2004,.

[3] A. Savvides, H. Park, and M.B. Srivastava, "The Bits and Flops of the N-hop Multilateration Primitive For Node Localization Problems," *ACM Conf. on Wireless Sensor Networks and Applications*, pp. 112-121, Sept. 2002.

[4] Y. Shang, W. Rumi, Y. Zhang, and M.P.J. Fromherz, "Localization from Mere Connectivity," *ACM Conf. on Mobile Ad Hoc Networking and Computing*, pp. 201-212, June 2003.

[5] S. Capkun, M. Hamdi, and J.-P. Hubaux, "GPS-free positioning in mobile Ad-Hoc networks," *IEEE Hawaii Conf. on System Sciences*, pp. 255-264, Jan. 2001.

[6] C. Savarese, J.M. Rabaey, and J. Beutel, "Location in Distributed Ad-Hoc Wireless Networks," *IEEE Conf. on Acoustics, Speech, and Signal Processing*, pp. 2037-2040, May 2001.

[7] L. Doherty, K.S.J. Pister, and L. El Ghaoui, "Convex Position Estimation in Wireless Sensor Networks," *IEEE Conf. on Information Theory and Communications*, pp. 1655-1663, April 2001.

[8] C. Gentile, "Sensor Location through Linear Programming with Triangle Inequality Constraints," *IEEE Conf. on Communications*, pp. 3192-3196, May 2005.

[9] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," *IEEE Computer*, vol. 34, no. 8, pp. 57-66, Aug. 2001.

[10] M.S. Bazaraa, J.J. Jarvis, and H.D. Sherali, "Linear Programming and Network Flows," *John Wiley & Sons, Inc.*, 1990.

[11] G.B. Dantzig, "Linear Programming and Extensions," *Princeton University Press*, 1963.

[12] S. Vajda, "Theory of Linear and Non-Linear Programming," *Longman Group Limited*, 1974.

[13] J. Werb and C. Lanzi, "Designing a postitioning system for finding things and people indoors," *IEEE Spectrum*, vol. 35, no. 9, pp. 71-78, Sept. 1998.

[14] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS)," *IEEE Conf. on Global Communications*, pp. 2926-2931, Nov. 2001.