by solving

$$\begin{array}{ll} \underset{A\in\mathbb{S}^{\mathbf{3}},\, B\in\mathbb{S}^{\mathbf{3}},\, W\in\mathbb{R}^{\mathbf{3}\times\mathbf{2}},\, H\in\mathbb{R}^{\mathbf{2}\times\mathbf{3}}}{\text{find}} & W\,,\ H \\ \text{subject to} & Z=\begin{bmatrix} I & W^{\mathrm{T}} & H \\ W & A & X \\ H^{\mathrm{T}} & X^{\mathrm{T}} & B \end{bmatrix}\succeq 0 \\ & W\geq \mathbf{0} \\ & H\geq \mathbf{0} \\ & \operatorname{rank} Z\leq 2 \end{array} \tag{809}$$

which follows from the fact, at optimality,

$$Z^{\star}=\begin{bmatrix} I \\ W \\ H^{\mathrm{T}} \end{bmatrix}\begin{bmatrix} I & W^{\mathrm{T}} & H \end{bmatrix} \tag{810}$$

Use the known closed-form solution for a direction vector Y to regulate rank by convex iteration; set $Z^{\star}=Q\Lambda Q^{\mathrm{T}}\in\mathbb{S}^{\mathbf{8}}$ to an ordered diagonalization and $U^{\star}=Q(:\,,3{:}8)\in\mathbb{R}^{\mathbf{8}\times\mathbf{6}}$, then $Y=U^{\star}U^{\star\mathrm{T}}$ (§4.5.1.1).

In summary, initialize Y then iterate numerical solution of (convex) semidefinite program

$$\begin{array}{ll} \underset{A\in\mathbb{S}^{\mathbf{3}},\, B\in\mathbb{S}^{\mathbf{3}},\, W\in\mathbb{R}^{\mathbf{3}\times\mathbf{2}},\, H\in\mathbb{R}^{\mathbf{2}\times\mathbf{3}}}{\text{minimize}} & \langle Z\,,\,Y\rangle \\ \text{subject to} & Z=\begin{bmatrix} I & W^{\mathrm{T}} & H \\ W & A & X \\ H^{\mathrm{T}} & X^{\mathrm{T}} & B \end{bmatrix}\succeq 0 \\ & W\geq \mathbf{0} \\ & H\geq \mathbf{0} \end{array} \tag{811}$$

with $Y=U^{\star}U^{\star\mathrm{T}}$ until convergence (which is to a global optimum, and occurs in very few iterations for this instance). ▼

Now, an application to optimal regulation of affine dimension:

4.5.1.2.4 Example. *Sensor-Network Localization and Wireless Location.*
Heuristic solution to a sensor-network localization problem, proposed by Carter, Jin, Saunders, & Ye in [77],[4.30] is limited to two Euclidean dimensions and applies semidefinite programming (SDP) to little subproblems. There, a large network is partitioned into smaller subnetworks (as small as one *sensor* - a mobile point, whereabouts unknown) and then semidefinite programming and heuristics called SPASELOC are applied to localize each and every partition by two-dimensional distance geometry. Their partitioning procedure is one-pass, yet termed *iterative*; a term applicable only insofar as adjoining partitions can share localized sensors and *anchors* (absolute sensor positions known *a priori*). But there is no iteration on the entire network, hence the term "iterative" is perhaps inappropriate. As partitions are selected based on "rule sets" (heuristics, not geographics), they also term the partitioning *adaptive.* But no adaptation of a partition actually occurs once it has been determined.

[4.30] The paper constitutes Jin's dissertation for University of Toronto [240] although her name appears as second author. Ye's authorship is honorary.

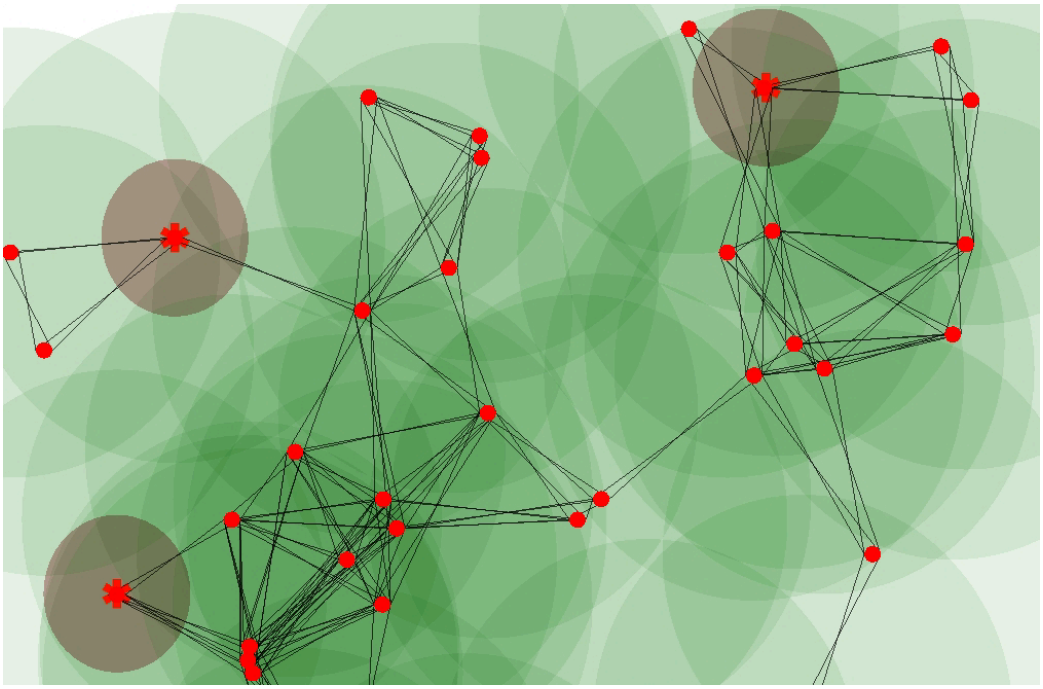

Figure 95: Sensor-network localization in $\mathbb{R}^2$, illustrating connectivity and circular radio-range per *sensor*. Smaller dark grey regions each hold an *anchor* at their center; known fixed sensor positions. Sensor/anchor distance is measurable with negligible uncertainty for sensor within those grey regions. (Graphic by Geoff Merrett.)

One can reasonably argue that semidefinite programming methods are unnecessary for localization of small partitions of large sensor networks. [306] [92] In the past, these nonlinear localization problems were solved algebraically and computed by least squares solution to hyperbolic equations; called *multilateration.*[4.31] [253] [293] Indeed, practical contemporary numerical methods for global positioning (GPS) by satellite do not rely on convex optimization. [319]

Modern distance geometry is inextricably melded with semidefinite programming. The beauty of semidefinite programming, as relates to localization, lies in convex expression of classical multilateration: So & Ye showed [353] that the problem of finding unique solution, to a noiseless nonlinear system describing the common point of intersection of hyperspheres in real Euclidean vector space, can be expressed as a semidefinite program via distance geometry.

But the need for SDP methods in Carter & Jin *et alii* is enigmatic for two more reasons: 1) guessing solution to a partition whose intersensor measurement data or connectivity is inadequate for localization by distance geometry, 2) reliance on complicated and extensive heuristics for partitioning a large network that could instead be efficiently solved whole by one semidefinite program [248, §3]. While partitions range in size between 2 and 10 sensors, 5 sensors optimally, heuristics provided are only for two spatial dimensions (no higher-dimensional heuristics are proposed). For these small numbers it remains unclarified as to precisely what advantage is gained over traditional

[4.31] Multilateration - literally, *having many sides*; shape of a geometric figure formed by nearly intersecting lines of position. In navigation systems, therefore: Obtaining a *fix* from multiple lines of position. Multilateration can be regarded as noisy trilateration.

least squares: it is difficult to determine what part of their noise performance is attributable to SDP and what part is attributable to their heuristic geometry.

Partitioning of large sensor networks is a compromise to rapid growth of SDP computational intensity with problem size. But when impact of noise on distance measurement is of most concern, one is averse to a partitioning scheme because noise-effects vary inversely with problem size. [56, §2.2] (§5.13.2) Since an individual partition's solution is not iterated in Carter & Jin and is interdependent with adjoining partitions, we expect errors to propagate from one partition to the next; the ultimate partition solved, expected to suffer most.

Heuristics often fail on real-world data because of unanticipated circumstances. When heuristics fail, generally they are repaired by adding more heuristics. Tenuous is any presumption, for example, that distance measurement errors have distribution characterized by circular contours of equal probability about an unknown sensor-location. (Figure **95**) That presumption effectively appears within Carter & Jin's optimization problem statement as affine equality constraints relating unknowns to distance measurements that are corrupted by noise. Yet in most all urban environments, this measurement noise is more aptly characterized by ellipsoids of varying orientation and eccentricity as one recedes from a sensor. (Figure **148**) Each unknown sensor must therefore instead be bound to its own particular range of distance, primarily determined by the terrain.[4.32] The nonconvex problem we must instead solve is:

$$\begin{array}{ll} \underset{i\,,j\,\in\,\mathcal{I}}{\text{find}} & \{x_i\ ,\ x_j\} \\ \text{subject to} & \underline{d_{ij}} \le \|x_i - x_j\|^2 \le \overline{d_{ij}} \end{array} \tag{812}$$

where x_i represents sensor location, and where $\underline{d_{ij}}$ and $\overline{d_{ij}}$ respectively represent lower and upper bounds on measured distance-square from i^{th} to j^{th} sensor (or from sensor to anchor). Figure **100** illustrates contours of equal sensor-location uncertainty. By establishing these individual upper and lower bounds, orientation and eccentricity can effectively be incorporated into the problem statement.

Generally speaking, there can be no unique solution to the sensor-network localization problem because there is no unique formulation; that is the art of Optimization. Any optimal solution obtained depends on whether or how a network is partitioned, whether distance data is complete, presence of noise, and how the problem is formulated. When a particular formulation is a convex optimization problem, then the set of all optimal solutions forms a convex set containing the actual or true localization. Measurement noise precludes equality constraints representing distance. The optimal solution set is consequently expanded; necessitated by introduction of distance inequalities admitting more and higher-rank solutions. Even were the optimal solution set a single point, it is not necessarily the true localization because there is little hope of exact localization by any algorithm once significant noise is introduced.

Carter & Jin gauge performance of their heuristics to the SDP formulation of author Biswas whom they regard as vanguard to the art. [15, §1] Biswas posed localization as an optimization problem minimizing a distance measure. [50] [48] Intuitively, minimization of any distance measure yields compacted solutions; (*confer* §6.7.0.0.1) precisely the anomaly motivating Carter & Jin. Their two-dimensional heuristics outperformed Biswas' localizations both in execution-time and proximity to the desired result. Perhaps, instead of heuristics, Biswas' approach to localization can be improved: [47] [49].

[4.32] A distinct contour map corresponding to each anchor is required in practice.

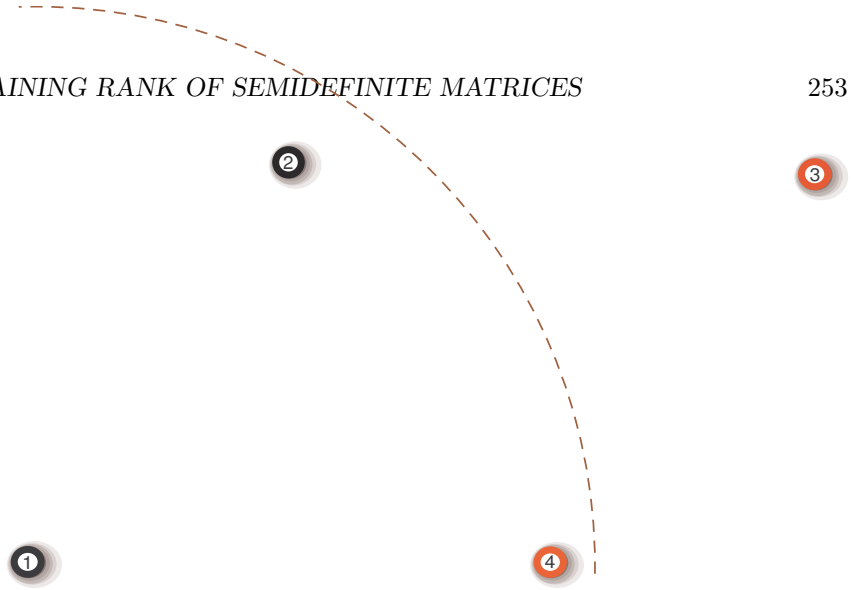

Figure 96: 2-lattice in $\mathbb{R}^2$, hand-drawn. Nodes 3 and 4 are anchors; remaining nodes are sensors. Radio range of sensor 1 indicated by arc.

The sensor-network localization problem is considered difficult. [15, §2] Rank constraints in optimization are considered more difficult. Control of affine dimension in Carter & Jin is suboptimal because of implicit projection on $\mathbb{R}^2$. In what follows, we present the localization problem as a semidefinite program (equivalent to (812)) having an explicit rank constraint which controls affine dimension of an optimal solution. We show how to achieve that rank constraint only if the feasible set contains a matrix of desired rank. Our problem formulation is extensible to any spatial dimension.

proposed standardized test

Jin proposes an academic test in two-dimensional real Euclidean space $\mathbb{R}^2$ that we adopt. In essence, this test is a localization of sensors and anchors arranged in a regular triangular lattice. Lattice connectivity is solely determined by sensor radio range; a connectivity graph is assumed incomplete. In the interest of test standardization, we propose adoption of a few small examples: Figure **96** through Figure **99** and their particular connectivity represented by matrices (813) through (816) respectively.

$$\begin{bmatrix} 0 & \bullet & ? & \bullet \\ \bullet & 0 & \bullet & \bullet \\ ? & \bullet & 0 & \circ \\ \bullet & \bullet & \circ & 0 \end{bmatrix} \tag{813}$$

Matrix entries *dot* $\bullet$ indicate measurable distance between *nodes* while unknown distance is denoted by ? (*question mark*). Matrix entries *hollow dot* $\circ$ represent known distance between anchors (to high accuracy) while zero distance is denoted 0. Because measured distances are quite unreliable in practice, our solution to the localization problem substitutes a distinct range of possible distance for each measurable distance; equality constraints exist only for anchors.

Anchors are chosen so as to increase difficulty for algorithms dependent on existence of sensors in their convex hull. The challenge is to find a solution in two dimensions close to the true sensor positions given incomplete noisy intersensor distance information.

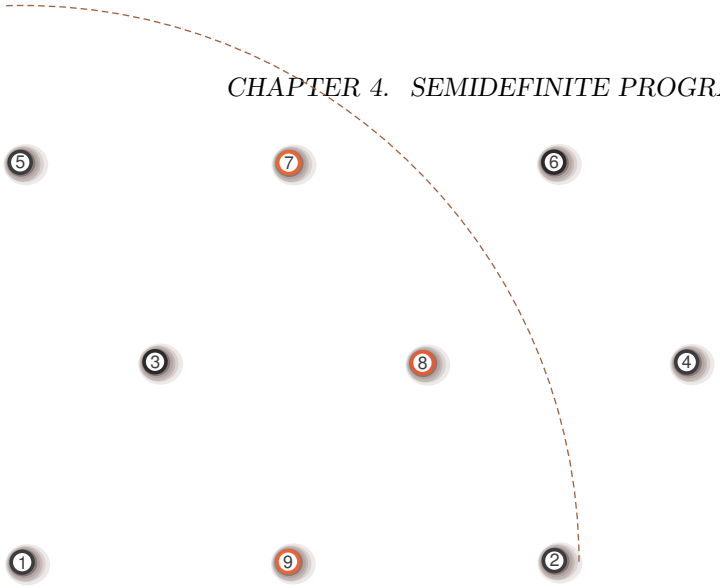

Figure 97: 3-lattice in $\mathbb{R}^2$, hand-drawn. Nodes 7, 8, and 9 are anchors; remaining nodes are sensors. Radio range of sensor 1 indicated by arc.

$$\begin{array}{ccccccccc}
0 & \bullet & \bullet & ? & \bullet & ? & ? & \bullet & \bullet \\
\bullet & 0 & \bullet & \bullet & ? & \bullet & ? & \bullet & \bullet \\
\bullet & \bullet & 0 & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
? & \bullet & \bullet & 0 & ? & \bullet & \bullet & \bullet & \bullet \\
\bullet & ? & \bullet & ? & 0 & \bullet & \bullet & \bullet & \bullet \\
? & \bullet & \bullet & \bullet & \bullet & 0 & \bullet & \bullet & \bullet \\
? & ? & \bullet & \bullet & \bullet & \bullet & 0 & \circ & \circ \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \circ & 0 & \circ \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \circ & \circ & 0
\end{array} \tag{814}$$

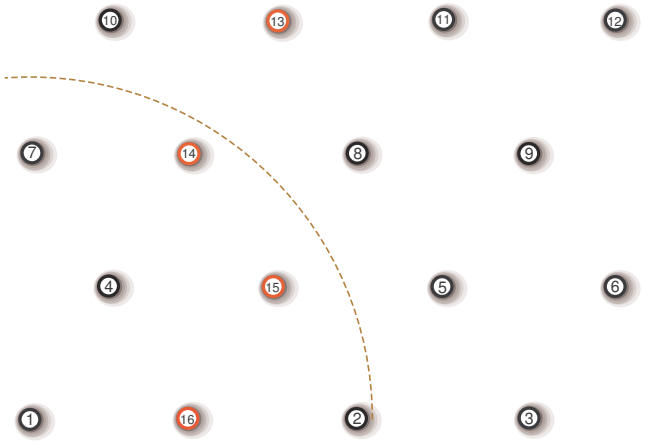

Figure 98: 4-lattice in $\mathbb{R}^2$, hand-drawn. Nodes 13, 14, 15, and 16 are anchors; remaining nodes are sensors. Radio range of sensor 1 indicated by arc.

$$
\begin{array}{cccccccccccccccc}
0 & ? & ? & \bullet & ? & ? & \bullet & ? & ? & ? & ? & ? & ? & ? & \bullet & \bullet \\
? & 0 & \bullet & \bullet & \bullet & \bullet & ? & \bullet & ? & ? & ? & ? & ? & & \bullet & \bullet \\
? & \bullet & 0 & ? & \bullet & \bullet & ? & ? & \bullet & ? & ? & ? & ? & ? & \bullet & \bullet \\
\bullet & \bullet & ? & 0 & \bullet & ? & \bullet & \bullet & ? & \bullet & ? & ? & \bullet & \bullet & \bullet & \bullet \\
? & \bullet & \bullet & \bullet & 0 & \bullet & ? & \bullet & \bullet & ? & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
? & \bullet & \bullet & ? & \bullet & 0 & ? & \bullet & \bullet & ? & \bullet & \bullet & ? & ? & ? & ? \\
\bullet & ? & ? & \bullet & ? & ? & 0 & ? & ? & \bullet & ? & ? & \bullet & \bullet & \bullet & \bullet \\
? & \bullet & ? & \bullet & \bullet & \bullet & ? & 0 & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
? & ? & \bullet & ? & \bullet & \bullet & ? & \bullet & 0 & ? & \bullet & \bullet & \bullet & ? & \bullet & ? \\
? & ? & ? & \bullet & ? & ? & \bullet & \bullet & ? & 0 & \bullet & ? & \bullet & \bullet & \bullet & ? \\
? & ? & ? & ? & \bullet & \bullet & ? & \bullet & \bullet & \bullet & 0 & \bullet & \bullet & \bullet & \bullet & ? \\
? & ? & ? & ? & \bullet & \bullet & ? & \bullet & \bullet & ? & \bullet & 0 & ? & ? & ? & ? \\
? & ? & ? & \bullet & \bullet & ? & \bullet & \bullet & \bullet & \bullet & \bullet & ? & 0 & \circ & \circ & \circ \\
? & \bullet & ? & \bullet & \bullet & ? & \bullet & \bullet & ? & \bullet & \bullet & ? & \circ & 0 & \circ & \circ \\
\bullet & \bullet & \bullet & \bullet & \bullet & ? & \bullet & \bullet & \bullet & \bullet & \bullet & ? & \circ & \circ & 0 & \circ \\
\bullet & \bullet & \bullet & \bullet & \bullet & ? & \bullet & \bullet & ? & ? & ? & ? & \circ & \circ & \circ & 0
\end{array}
\tag{815}
$$

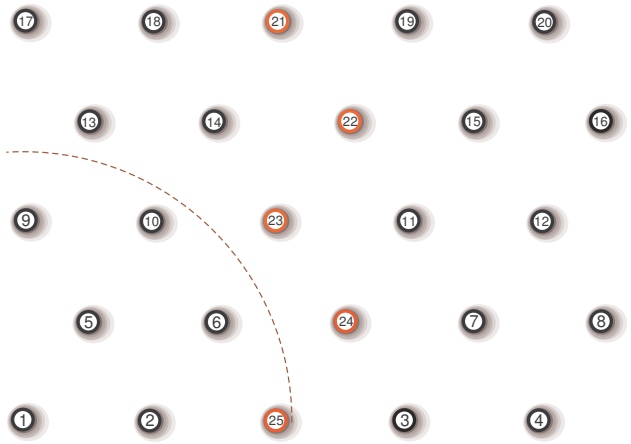

Figure 99: 5-lattice in $\mathbb{R}^2$. Nodes 21 through 25 are anchors.

$$
\begin{array}{cccccccccccccccccccccccc}
0 & \bullet & ? & ? & \bullet & \bullet & ? & ? & \bullet & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\
\bullet & 0 & ? & ? & \bullet & \bullet & ? & ? & ? & \bullet & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & \bullet & \bullet \\
? & ? & 0 & \bullet & ? & \bullet & \bullet & \bullet & ? & ? & \bullet & \bullet & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & \bullet & \bullet & \bullet \\
? & ? & \bullet & 0 & ? & ? & \bullet & \bullet & ? & ? & ? & \bullet & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & \bullet & ? \\
\bullet & \bullet & ? & ? & 0 & \bullet & ? & ? & \bullet & \bullet & ? & ? & \bullet & \bullet & ? & ? & \bullet & ? & ? & ? & ? & ? & \bullet & ? & \bullet \\
\bullet & \bullet & \bullet & ? & \bullet & 0 & \bullet & ? & \bullet & \bullet & \bullet & ? & ? & \bullet & ? & ? & ? & ? & ? & ? & ? & ? & \bullet & \bullet & \bullet \\
? & ? & \bullet & \bullet & ? & \bullet & 0 & \bullet & ? & ? & \bullet & \bullet & ? & ? & \bullet & \bullet & ? & ? & ? & ? & ? & ? & \bullet & \bullet & \bullet \\
? & ? & \bullet & \bullet & ? & ? & \bullet & 0 & ? & ? & \bullet & \bullet & ? & ? & \bullet & \bullet & ? & ? & ? & ? & ? & ? & \bullet & ? \\
\bullet & ? & ? & ? & \bullet & \bullet & ? & ? & 0 & \bullet & ? & ? & \bullet & \bullet & ? & ? & \bullet & \bullet & ? & ? & ? & ? & ? & ? \\
? & \bullet & ? & ? & \bullet & \bullet & ? & ? & \bullet & 0 & \bullet & ? & \bullet & \bullet & ? & ? & ? & \bullet & ? & ? & \bullet & \bullet & \bullet & \bullet \\
? & ? & \bullet & ? & ? & \bullet & \bullet & \bullet & ? & \bullet & 0 & \bullet & ? & \bullet & \bullet & \bullet & ? & ? & \bullet & ? & ? & \bullet & \bullet & \bullet \\
? & ? & \bullet & \bullet & ? & ? & \bullet & \bullet & ? & ? & \bullet & 0 & ? & ? & \bullet & \bullet & ? & ? & \bullet & \bullet & ? & \bullet & \bullet & \bullet & ? \\
? & ? & ? & ? & \bullet & ? & ? & ? & \bullet & \bullet & ? & ? & 0 & \bullet & ? & ? & \bullet & \bullet & ? & ? & \bullet & \bullet & ? & ? & ? \\
? & ? & ? & ? & \bullet & \bullet & ? & ? & \bullet & \bullet & \bullet & ? & \bullet & 0 & \bullet & ? & \bullet & \bullet & \bullet & ? & \bullet & \bullet & \bullet & \bullet & ? \\
? & ? & ? & ? & ? & ? & \bullet & \bullet & ? & ? & \bullet & \bullet & ? & \bullet & 0 & \bullet & ? & ? & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & ? \\
? & ? & ? & ? & ? & ? & \bullet & \bullet & ? & ? & \bullet & \bullet & ? & ? & \bullet & 0 & ? & ? & \bullet & \bullet & ? & \bullet & ? & ? & ? \\
? & ? & ? & ? & \bullet & ? & ? & ? & \bullet & ? & ? & ? & \bullet & \bullet & ? & ? & 0 & \bullet & ? & ? & \bullet & ? & ? & ? & ? \\
? & ? & ? & ? & ? & ? & ? & ? & \bullet & \bullet & ? & ? & \bullet & \bullet & ? & ? & \bullet & 0 & \bullet & ? & \bullet & \bullet & \bullet & ? & ? \\
? & ? & ? & ? & ? & ? & ? & ? & ? & ? & \bullet & \bullet & ? & \bullet & \bullet & \bullet & ? & \bullet & 0 & \bullet & \bullet & \bullet & \bullet & ? & ? \\
? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & \bullet & ? & ? & \bullet & \bullet & ? & ? & \bullet & 0 & \bullet & \bullet & ? & ? & ? \\
? & ? & ? & ? & ? & ? & ? & ? & ? & \bullet & ? & ? & \bullet & \bullet & \bullet & ? & \bullet & \bullet & \bullet & \bullet & 0 & \circ & \circ & \circ & \circ \\
? & ? & ? & ? & ? & ? & ? & ? & ? & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & ? & \bullet & \bullet & \bullet & \circ & 0 & \circ & \circ & \circ \\
? & ? & \bullet & ? & \bullet & \bullet & \bullet & ? & ? & \bullet & \bullet & \bullet & ? & \bullet & \bullet & ? & ? & \bullet & \bullet & ? & \circ & \circ & 0 & \circ & \circ \\
? & \bullet & \bullet & \bullet & ? & \bullet & \bullet & \bullet & ? & \bullet & \bullet & \bullet & ? & \bullet & \bullet & ? & ? & ? & ? & ? & \circ & \circ & \circ & 0 & \circ \\
? & \bullet & \bullet & ? & \bullet & \bullet & \bullet & ? & ? & \bullet & \bullet & ? & ? & ? & ? & ? & ? & ? & ? & ? & \circ & \circ & \circ & \circ & 0
\end{array}
\tag{816}
$$

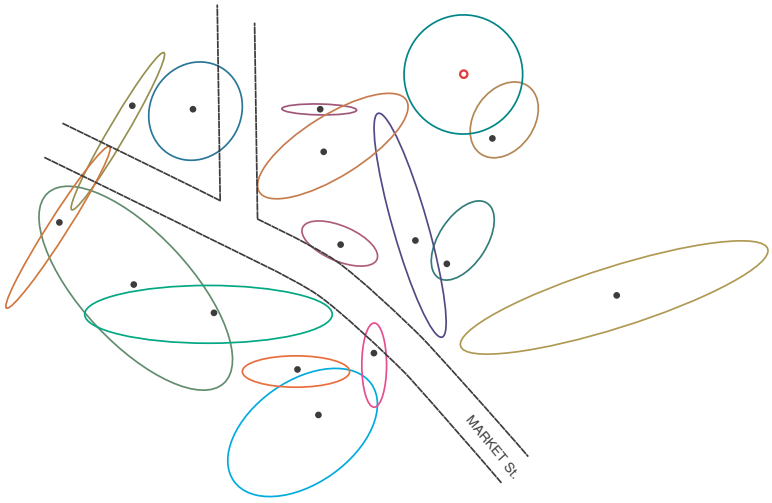

Figure 100: Location uncertainty ellipsoid in $\mathbb{R}^2$ for each of 15 sensors $\bullet$ within three city blocks in downtown San Francisco. (Data by Polaris Wireless.)

problem statement

Ascribe points in a list $\{x_\ell \in \mathbb{R}^n,\ \ell = 1 \dots N\}$ to the columns of a matrix X;

$$X = [\, x_1 \ \cdots \ x_N \,] \in \mathbb{R}^{n \times N} \qquad (77)$$

where N is regarded as cardinality of list X. Positive semidefinite matrix $X^{\mathrm{T}}X$, formed from inner product of the list, is a *Gram matrix*; [277, §3.6]

$$G = X^{\mathrm{T}}X = \begin{bmatrix} \|x_1\|^2 & x_1^{\mathrm{T}}x_2 & x_1^{\mathrm{T}}x_3 & \cdots & x_1^{\mathrm{T}}x_N \\ x_2^{\mathrm{T}}x_1 & \|x_2\|^2 & x_2^{\mathrm{T}}x_3 & \cdots & x_2^{\mathrm{T}}x_N \\ x_3^{\mathrm{T}}x_1 & x_3^{\mathrm{T}}x_2 & \|x_3\|^2 & \ddots & x_3^{\mathrm{T}}x_N \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ x_N^{\mathrm{T}}x_1 & x_N^{\mathrm{T}}x_2 & x_N^{\mathrm{T}}x_3 & \cdots & \|x_N\|^2 \end{bmatrix} \in \mathbb{S}_+^N \qquad (1038)$$

where $\mathbb{S}_+^N$ is the convex cone of $N \times N$ positive semidefinite matrices in the symmetric matrix subspace $\mathbb{S}^N$.

Existence of noise precludes measured distance from the input data. We instead assign measured distance to a range estimate specified by individual upper and lower bounds: $\overline{d_{ij}}$ is an upper bound on distance-square from i^{th} to j^{th} sensor, while $\underline{d_{ij}}$ is a lower bound. These bounds become the input data. Each measurement range is presumed different from the others because of measurement uncertainty; *e.g*, Figure **100**.

Our mathematical treatment of anchors and sensors is not dichotomized.[4.33] A sensor position that is known *a priori* to high accuracy (with absolute certainty) $\check{x}_i$ is called an *anchor*. Then the sensor-network localization problem (812) can be expressed equivalently: Given a number m of anchors and a set of indices $\mathcal{I}$ (corresponding to all measurable distances $\bullet$), for $0 < n < N$

[4.33] *Wireless location* problem thus stated identically; difference being: fewer sensors.

$$\begin{array}{ll}\underset{G\in\mathbb{S}^N,\ X\in\mathbb{R}^{n\times N}}{\text{find}} & X \\ \text{subject to} & \underline{d_{ij}} \le \langle G\ ,\ (e_i-e_j)(e_i-e_j)^{\mathrm{T}}\rangle \le \overline{d_{ij}} \qquad \forall (i,j)\in\mathcal{I} \\ & \langle G\ ,\ e_ie_i^{\mathrm{T}}\rangle = \|\check{x}_i\|^2\ , \quad i=N-m+1\dots N \\ & \langle G\ ,\ (e_ie_j^{\mathrm{T}}+e_je_i^{\mathrm{T}})/2\rangle = \check{x}_i^{\mathrm{T}}\check{x}_j\ , \quad i<j\ , \quad \forall i,j\in\{N-m+1\dots N\} \\ & X(:\,,N-m+1{:}N) = [\,\check{x}_{N-m+1}\ \cdots\ \check{x}_N\,] \\ & Z=\begin{bmatrix} I & X \\ X^{\mathrm{T}} & G\end{bmatrix} \succeq 0 \\ & \operatorname{rank} Z = n \end{array} \tag{817}$$

where e_i is the i^{th} member of the standard basis for $\mathbb{R}^N$. Distance-square

$$d_{ij} = \|x_i-x_j\|_2^2 = \langle x_i-x_j\ ,\ x_i-x_j\rangle \tag{1025}$$

is related to Gram matrix entries $G\triangleq[g_{ij}]$ by vector inner-product

$$\begin{aligned} d_{ij} &= g_{ii}+g_{jj}-2g_{ij} \\ &= \langle G\ ,\ (e_i-e_j)(e_i-e_j)^{\mathrm{T}}\rangle = \operatorname{tr}(G^{\mathrm{T}}(e_i-e_j)(e_i-e_j)^{\mathrm{T}}) \end{aligned} \tag{1040}$$

hence the scalar inequalities. Each linear equality constraint in $G\in\mathbb{S}^N$ represents a hyperplane in isometrically isomorphic Euclidean vector space $\mathbb{R}^{N(N+1)/2}$, while each linear inequality pair represents a convex Euclidean body known as *slab*.[4.34] By Schur complement (§A.4), any solution $(G\,,X)$ provides comparison with respect to the positive semidefinite cone

$$G \succeq X^{\mathrm{T}}X \tag{1078}$$

which is a convex relaxation of the desired equality constraint

$$\begin{bmatrix} I & X \\ X^{\mathrm{T}} & G\end{bmatrix} = \begin{bmatrix} I \\ X^{\mathrm{T}}\end{bmatrix}\begin{bmatrix} I & X\end{bmatrix} \tag{1079}$$

The rank constraint insures this equality holds, by Theorem A.4.0.1.3, thus restricting solution to $\mathbb{R}^n$. Assuming full-rank solution (list) X

$$\operatorname{rank} Z = \operatorname{rank} G = \operatorname{rank} X \tag{818}$$

convex equivalent problem statement

Problem statement (817) is nonconvex because of the rank constraint. We do not eliminate or ignore the rank constraint; rather, we find a convex way to enforce it: for $0<n<N$

$$\begin{array}{ll}\underset{G\in\mathbb{S}^N,\ X\in\mathbb{R}^{n\times N}}{\text{minimize}} & \langle Z\ ,\ W\rangle \\ \text{subject to} & \underline{d_{ij}} \le \langle G\ ,\ (e_i-e_j)(e_i-e_j)^{\mathrm{T}}\rangle \le \overline{d_{ij}} \qquad \forall (i,j)\in\mathcal{I} \\ & \langle G\ ,\ e_ie_i^{\mathrm{T}}\rangle = \|\check{x}_i\|^2\ , \quad i=N-m+1\dots N \\ & \langle G\ ,\ (e_ie_j^{\mathrm{T}}+e_je_i^{\mathrm{T}})/2\rangle = \check{x}_i^{\mathrm{T}}\check{x}_j\ , \quad i<j\ , \quad \forall i,j\in\{N-m+1\dots N\} \\ & X(:\,,N-m+1{:}N) = [\,\check{x}_{N-m+1}\ \cdots\ \check{x}_N\,] \\ & Z=\begin{bmatrix} I & X \\ X^{\mathrm{T}} & G\end{bmatrix} \succeq 0 \end{array} \tag{819}$$

[4.34] an intersection of two parallel but opposing halfspaces (Figure 13). In terms of position X, this distance slab can be thought of as a thick *hypershell* instead of a hypersphere boundary.

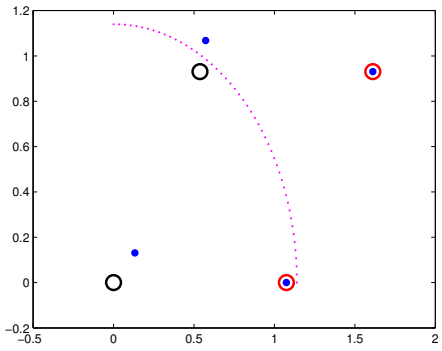

Figure 101: Typical solution for 2-lattice in Figure **96** with noise factor $\eta = 0.1$ (822). Two red rightmost nodes are anchors; two remaining nodes are sensors. Radio range of sensor 1 indicated by arc; radius = 1.14. Actual sensor indicated by target ○ while its localization is indicated by bullet • . Rank-2 solution found in 1 iteration (819) (1866a) subject to reflection error.

Convex function $\operatorname{tr} Z$ is a well-known heuristic whose sole purpose is to represent convex envelope of $\operatorname{rank} Z$. (§7.2.2.1) In this convex optimization problem (819), a semidefinite program, we substitute a vector inner-product objective function for trace;

$$\operatorname{tr} Z \;=\; \langle Z\,,\, I\rangle \;\leftarrow\; \langle Z\,,\, W\rangle \tag{820}$$

a generalization of the trace heuristic for minimizing convex envelope of rank, where $W \in \mathbb{S}_+^{N+n}$ is constant with respect to (819). Matrix W is normal to a hyperplane in $\mathbb{S}^{N+n}$ minimized over a convex feasible set specified by the constraints in (819). Matrix W is chosen so $-W$ points in direction of rank-n feasible solutions G. For properly chosen W, problem (819) becomes an equivalent to (817). Thus the purpose of vector inner-product objective (820) is to locate a rank-n feasible Gram matrix assumed existent on the boundary of positive semidefinite cone $\mathbb{S}_+^N$, as explained beginning in §4.5.1; how to choose direction vector W is explained there and in what follows:

direction matrix W

Denote by $Z^\star$ an optimal composite matrix from semidefinite program (819). Then for $Z^\star \in \mathbb{S}^{N+n}$ whose eigenvalues $\lambda(Z^\star) \in \mathbb{R}^{N+n}$ are arranged in nonincreasing order, (Ky Fan)

$$\sum_{i=n+1}^{N+n} \lambda(Z^\star)_i \;=\; \begin{array}{ll} \underset{W\in\mathbb{S}^{N+n}}{\text{minimize}} & \langle Z^\star,\, W\rangle \\ \text{subject to} & 0 \preceq W \preceq I \\ & \operatorname{tr} W = N \end{array} \tag{1866a}$$

which has an optimal solution that is known in closed form (p.529, §4.5.1.1). This eigenvalue sum is zero when $Z^\star$ has rank n or less.

Foreknowledge of optimal $Z^\star$, to make possible this search for W, implies iteration; *id est*, semidefinite program (819) is solved for $Z^\star$ initializing $W = I$ or $W = \mathbf{0}$. Once found, $Z^\star$ becomes constant in semidefinite program (1866a) where a new normal direction W is found as its optimal solution. Then this cycle (819) (1866a) iterates until convergence.

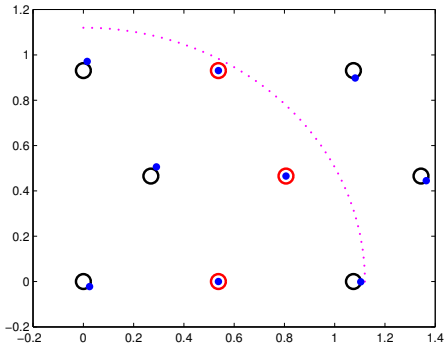

Figure 102: Typical solution for 3-lattice in Figure **97** with noise factor $\eta = 0.1$ (822). Three red vertical middle nodes are anchors; remaining nodes are sensors. Radio range of sensor 1 indicated by arc; radius $= 1.12$. Actual sensor indicated by target $\circ$ while its localization is indicated by bullet $\bullet$. Rank-2 solution found in 2 iterations (819) (1866a).

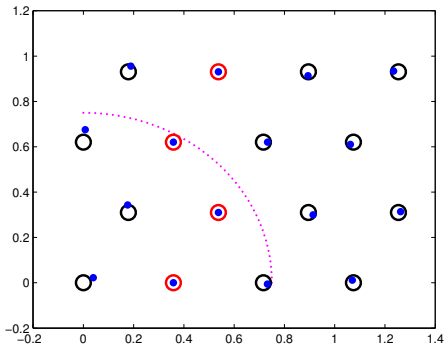

Figure 103: Typical solution for 4-lattice in Figure **98** with noise factor $\eta = 0.1$ (822). Four red vertical middle-left nodes are anchors; remaining nodes are sensors. Radio range of sensor 1 indicated by arc; radius $= 0.75$. Actual sensor indicated by target $\circ$ while its localization is indicated by bullet $\bullet$. Rank-2 solution found in 7 iterations (819) (1866a).

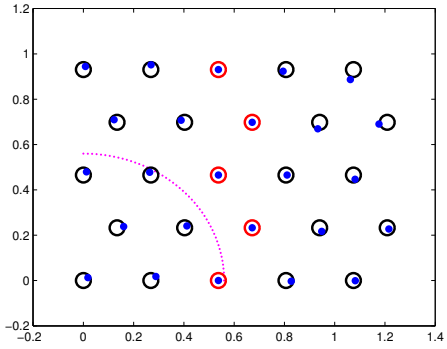

Figure 104: Typical solution for 5-lattice in Figure **99** with noise factor $\eta = 0.1$ (822). Five red vertical middle nodes are anchors; remaining nodes are sensors. Radio range of sensor 1 indicated by arc; radius $= 0.56$. Actual sensor indicated by target $\circ$ while its localization is indicated by bullet $\bullet$. Rank-2 solution found in 3 iterations (819) (1866a).

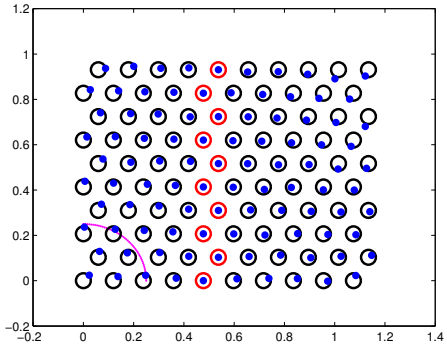

Figure 105: Typical solution for 10-lattice with noise factor $\eta = 0.1$ (822) compares better than Carter & Jin [77, fig.4.2]. Ten red vertical middle nodes are anchors; the rest are sensors. Radio range of sensor 1 indicated by arc; radius $= 0.25$. Actual sensor indicated by target $\circ$ while its localization is indicated by bullet $\bullet$. Rank-2 solution found in 5 iterations (819) (1866a).

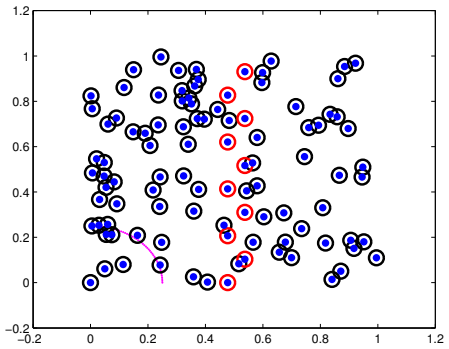

Figure 106: Typical localization of 100 randomized noiseless sensors ($\eta = 0$ (822)) is exact despite incomplete EDM. Ten red vertical middle nodes are anchors; remaining nodes are sensors. Radio range of sensor at origin indicated by arc; radius $= 0.25$. Actual sensor indicated by target $\bigcirc$ while its localization is indicated by bullet $\bullet$. Rank-2 solution found in 3 iterations (819) (1866a).

When $\operatorname{rank} Z^\star = n$, solution via this convex iteration solves sensor-network localization problem (812) and its equivalent (817).

numerical solution

In all examples to follow, number of anchors

$$m = \sqrt{N} \tag{821}$$

equals square root of cardinality N of list X. Indices set $\mathcal{I}$ identifying all measurable distances $\bullet$ is ascertained from connectivity matrix (813), (814), (815), or (816). We solve iteration (819) (1866a) in dimension $n = 2$ for each respective example illustrated in Figure **96** through Figure **99**.

In presence of negligible noise, true position is reliably localized for every standardized example; noteworthy insofar as each example represents an incomplete graph. This implies that the set of all optimal solutions having least rank must be small.

To make the examples interesting and consistent with previous work, we randomize each range of distance-square that bounds $\langle G\,,\,(e_i - e_j)(e_i - e_j)^{\mathrm{T}} \rangle$ in (819); *id est*, for each and every $(i\,,j) \in \mathcal{I}$

$$\begin{aligned} \overline{d_{ij}} &= d_{ij}(1 + \sqrt{3}\,\eta\,\chi_l)^2 \\ \underline{d_{ij}} &= d_{ij}(1 - \sqrt{3}\,\eta\,\chi_{l+1})^2 \end{aligned} \tag{822}$$

where $\eta = 0.1$ is a constant noise factor, χ_l is the l^{th} sample of a noise process realization uniformly distributed in the interval $(0\,,1)$ like `rand(1)` from MATLAB, and d_{ij} is actual distance-square from i^{th} to j^{th} sensor. Because of distinct function calls to `rand()`, each range of distance-square $[\,\underline{d_{ij}}\,,\,\overline{d_{ij}}\,]$ is not necessarily centered on actual distance-square d_{ij}. Unit stochastic variance is provided by factor $\sqrt{3}$.

Figure **101** through Figure **104** each illustrate one realization of numerical solution to the standardized lattice problems posed by Figure **96** through Figure **99** respectively. Exact localization, by any method, is impossible because of measurement noise. Certainly, by inspection of their published graphical data, our results are better than those of Carter & Jin. (Figure **105**, **106**, **107**) Obviously our solutions do not suffer from those

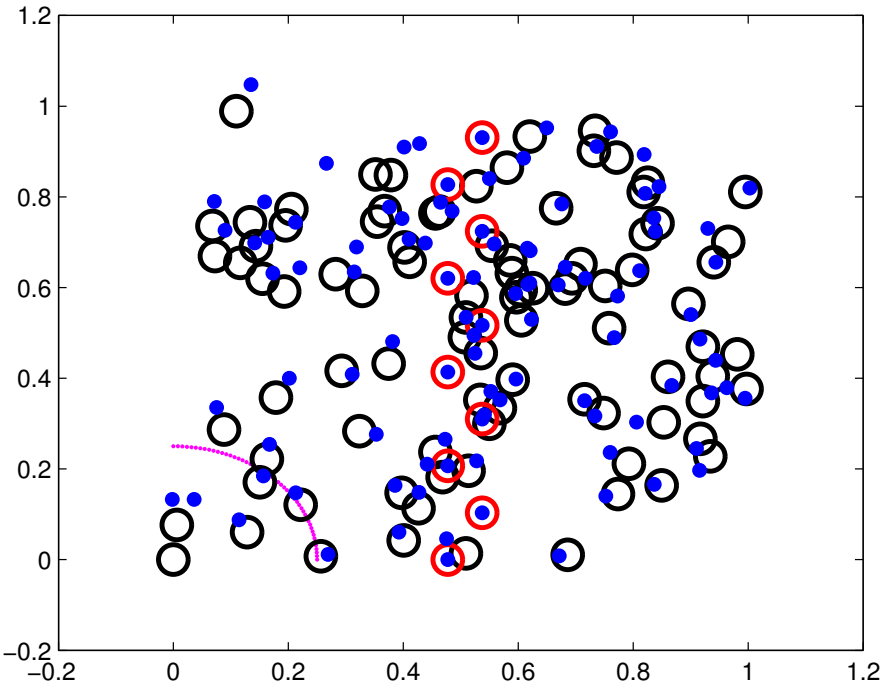

Figure 107: Typical solution for 100 randomized sensors with noise factor $\eta = 0.1$ (822); worst measured average sensor error ≈ 0.0044 compares better than Carter & Jin's 0.0154 computed in 0.71s [77, p.19]. Ten red vertical middle nodes are anchors; same as before. Remaining nodes are sensors. Interior anchor placement makes localization difficult. Radio range of sensor at origin indicated by arc; radius $= 0.25$. Actual sensor indicated by target ○ while its localization is indicated by bullet •. After 1 iteration $\operatorname{rank} G = 92$, after 2 iterations $\operatorname{rank} G = 4$. Rank-2 solution found in 3 iterations (819) (1866a). (Regular lattice in Figure **105** is actually harder to solve, requiring more iterations.) Runtime for SDPT3 [384] under `cvx` [188] is a few minutes on 2009 vintage laptop Core 2 Duo CPU (Intel T6400@2GHz, 800MHz FSB).

compaction-type errors (clustering of localized sensors) exhibited by Biswas' graphical results for the same noise factor η.

localization example conclusion

Solution to this sensor-network localization problem became apparent by understanding geometry of optimization. Trace of a matrix, to a student of linear algebra, is perhaps a sum of eigenvalues. But to us, trace represents the normal I to some hyperplane in Euclidean vector space. (Figure **94**)

Our solutions are globally optimal, requiring: 1) no centralized-gradient postprocessing heuristic refinement as in [47] because there is effectively no relaxation of (817) at global optimality, 2) no implicit postprojection on rank-2 positive semidefinite matrices induced by nonzero $G-X^{\mathrm{T}}X$ denoting suboptimality as occurs in [48] [49] [50] [77] [240] [248]; indeed, $G^\star = X^{\star\mathrm{T}}X^\star$ by convex iteration.

Numerical solution to noisy problems, containing sensor variables well in excess of 100, becomes difficult via the holistic semidefinite program we proposed. When problem size is within reach of contemporary general-purpose semidefinite program solvers, then the convex iteration we presented inherently overcomes limitations of Carter & Jin with respect to both noise performance and ability to localize in any desired affine dimension.

The legacy of Carter, Jin, Saunders, & Ye [77] is a sobering demonstration of the need for more efficient methods for solution of semidefinite programs, while that of So & Ye [353] forever bonds distance geometry to semidefinite programming. Elegance of our semidefinite problem statement (819), for constraining affine dimension of sensor-network localization, should provide some *impetus* to focus more research on computational intensity of general-purpose semidefinite program solvers. An approach different from interior-point methods is required; higher speed and greater accuracy from a simplex-like solver is what is needed. □

4.5.1.2.5 Example. *Nonnegative spectral factorization.* (*confer* §3.14.1.0.2)
Having found optimal real coefficient vectors $v^\star, u^\star$ for a sixteenth order magnitude square transfer function, evaluated along the $\jmath\omega$ axis (p.209),

$$|H(\jmath\omega)|^2 = H(\jmath\omega)H(-\jmath\omega) = \frac{1 + v_1^\star\omega^2 + v_2^\star\omega^4 + \ldots + v_8^\star\omega^{16}}{1 + u_1^\star\omega^2 + u_2^\star\omega^4 + \ldots + u_8^\star\omega^{16}} \qquad (667)$$

we wish to find real coefficients b, a for corresponding Fourier transform

$$H(\jmath\omega) = \frac{1 + b_1\jmath\omega + b_2(\jmath\omega)^2 + \ldots + b_8(\jmath\omega)^8}{1 + a_1\jmath\omega + a_2(\jmath\omega)^2 + \ldots + a_8(\jmath\omega)^8} \qquad (664)$$

These coefficients $b, a, v^\star, u^\star$ are related through simultaneous nonlinear algebraic equations:

$$\begin{aligned}
v_1^\star &= b_1^2 - 2b_2 \\
v_2^\star &= b_2^2 - 2b_1b_3 + 2b_4 \\
v_3^\star &= b_3^2 - 2b_2b_4 + 2b_1b_5 - 2b_6 \\
v_4^\star &= b_4^2 - 2b_3b_5 + 2b_2b_6 - 2b_1b_7 + 2b_8 \\
v_5^\star &= b_5^2 - 2b_4b_6 + 2b_3b_7 - 2b_2b_8 \\
v_6^\star &= b_6^2 - 2b_5b_7 + 2b_4b_8 \\
v_7^\star &= b_7^2 - 2b_6b_8 \\
v_8^\star &= b_8^2
\end{aligned}
\qquad
\begin{aligned}
u_1^\star &= a_1^2 - 2a_2 \\
u_2^\star &= a_2^2 - 2a_1a_3 + 2a_4 \\
u_3^\star &= a_3^2 - 2a_2a_4 + 2a_1a_5 - 2a_6 \\
u_4^\star &= a_4^2 - 2a_3a_5 + 2a_2a_6 - 2a_1a_7 + 2a_8 \\
u_5^\star &= a_5^2 - 2a_4a_6 + 2a_3a_7 - 2a_2a_8 \\
u_6^\star &= a_6^2 - 2a_5a_7 + 2a_4a_8 \\
u_7^\star &= a_7^2 - 2a_6a_8 \\
u_8^\star &= a_8^2
\end{aligned}
\qquad (823)$$