Exploiting sparsity in polynomial optimization problems

NONCONVEX PROGRAMMING: LOCAL and GLOBAL APPROACHES

Theory, Algorithms and Applications

National Institute for Applied Sciences, Rouen, France
17-21 December, 2007

Masakazu Kojima

Tokyo Institute of Technology

Contents

1. Polynomial Optimization Problems (POPs)
2. Semidefinite Programming (SDP) relaxations of POPs
3. How do we formulate structured sparsity?
4. Sparse SDP relaxations of POPs — briefly
5. Exploiting free variables in primal-dual interior-point methods for LP, SDP and SOCP
6. Application to sensor network localization problems
7. Concluding remarks

Contents

1. Polynomial Optimization Problems (POPs)
2. Semidefinite Programming (SDP) relaxations of POPs
3. How do we formulate structured sparsity?
4. Sparse SDP relaxations of POPs — briefly
5. Exploiting free variables in primal-dual interior-point methods for LP, SDP and SOCP
6. Application to sensor network localization problems
7. Concluding remarks

Notation and Symbols

$\mathbb{R}^n$: the n-dim Euclidean space.

$\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$: a vector variable.

$f_j(\boldsymbol{x})$: a multivariate polynomial in $\boldsymbol{x} \in \mathbb{R}^n$ $(j = 0, 1, \ldots, m)$.

$$\boxed{\text{POP: min } f_0(\boldsymbol{x}) \text{ sub.to } f_j(\boldsymbol{x}) \geq 0 \text{ or } = 0 \ (j = 1, \ldots, m).}$$

Example: $n = 3$

$$\begin{array}{rl}
\text{min} & f_0(\boldsymbol{x}) \equiv x_1^3 - 2x_1x_2^2 + x_1^2x_2x_3 - 4x_3^2 \\
\text{sub.to} & f_1(\boldsymbol{x}) \equiv -x_1^2 + 5x_2x_3 + 1 \geq 0, \\
 & f_2(\boldsymbol{x}) \equiv x_1^2 - 3x_1x_2x_3 + 2x_3 + 2 \geq 0, \\
 & f_3(\boldsymbol{x}) \equiv -x_1^2 - x_2^2 - x_3^2 + 1 \geq 0, \\
 & x_1(x_1 - 1) = 0 \text{ (0-1 integer)}, \\
 & x_2 \geq 0, \ x_3 \geq 0, \ x_2x_3 = 0 \text{ (complementarity)}.
\end{array}$$

Contents

$$\text{POP:} \quad \min \quad f_0(\boldsymbol{x}) \quad \text{sub.to} \quad f_j(\boldsymbol{x}) \geq 0 \ (j = 1, \ldots, m).$$

[1] Lasserre, “Global optimization with polynomials and the problems of moments”, *SIAM J. on Optim.* (2001).
[2] Parrilo, “Semidefinite programming relaxations for semialgebraic problems”, *Math. Prog.* (2003).

- primal approach $\Rightarrow$ a sequence of SDP relaxations.
- dual approach $\Rightarrow$ a sequence of SOS relaxations.

$$\boxed{\text{POP:} \quad \min \quad f_0(\boldsymbol{x}) \quad \text{sub.to} \quad f_j(\boldsymbol{x}) \geq 0 \ (j = 1, \ldots, m).}$$

[1] Lasserre, "Global optimization with polynomials and the problems of moments", *SIAM J. on Optim.* (2001).

[2] Parrilo, "Semidefinite programming relaxations for semialgebraic problems", *Math. Prog.* (2003).

- primal approach $\Rightarrow$ a sequence of SDP relaxations.
- dual approach $\Rightarrow$ a sequence of SOS relaxations.

Main features:

(a) Lower bounds for the optimal value.

(b) Convergence to global optimal solutions under assump.

(c) Each relaxed problem can be solved as an SDP; its size $\uparrow$ rapidly along "the sequence" as the size of POP $\uparrow$, the deg. of poly. $\uparrow$, and/or we require higher accuracy.

(d) Expensive to solve large scale POPs in practice.

$\Rightarrow$ Exploiting Sparsity.

POP: $\quad \min \quad f_0(\boldsymbol{x}) \quad$ sub.to $\quad f_j(\boldsymbol{x}) \geq 0 \ (j = 1, \ldots, m).$

POP: $\min \quad f_0(\boldsymbol{x})$ sub.to $f_j(\boldsymbol{x}) \geq 0 \ (j = 1, \ldots, m)$.

Exploiting sparsity to solve larger scale problem in practice

[3] Kobayashi-Kim-Kojima, “Correlative sparsity in primal-dual interior-point methods for LP, SDP and SOCP”, Sep. 2006
$\Rightarrow$ Section 3

[4] Waki-Kim-Kojima-Muramatsu, “SOS and SDP relaxations for POPs with Structured Sparsity", *SIAM J. on Optim* (2006)
$\Rightarrow$ Section 4

Exploiting equalities in dual (free variables in primal) SDPs

[5] Kobayashi-Nakata-Kojima, "A Conversion of an SDP Having free variables into the Standard Form SDP", *Comp. Optim. Appl.* (2007)
$\Rightarrow$ Section 5

$\Rightarrow$ Appl. to sensor network localization problems in Section 6

Contents

1. Polynomial Optimization Problems (POPs)
2. Semidefinite Programming (SDP) relaxations of POPs
3. How do we formulate structured sparsity?
4. Sparse SDP relaxations of POPs — briefly
5. Exploiting free variables in primal-dual interior-point methods for LP, SDP and SOCP
6. Application to sensor network localization problems
7. Concluding remarks

POP: $\min \quad f_0(\boldsymbol{x}) \quad$ sub.to $\quad f_j(\boldsymbol{x}) \geq 0 \; (j = 1, \ldots, m).$

How do we exploit sparsity in POP?

$$\Downarrow$$

The answer depends on which methods we use to solve POP.

POP

$\Downarrow$ SDP relaxation (Lasserre 2001)

SDP $\Leftarrow$ Primal-Dual IPM (Interior-Point Method)

POP: $\min \quad f_0(\boldsymbol{x}) \quad$ sub.to $\quad f_j(\boldsymbol{x}) \geq 0 \ (j = 1, \ldots, m).$

How do we exploit sparsity in POP?

$\Downarrow$

The answer depends on which methods we use to solve POP.

POP

$\Downarrow$ SDP relaxation (Lasserre 2001)

SDP $\Leftarrow$ Primal-Dual IPM (Interior-Point Method)

We will assume a structured sparsity (correlative sparsity):

(a) A sparse SDP relaxation $\Rightarrow$ SDP of smaller size.

(b) SDP satisfies "a similar structured sparsity" under which Primal-Dual IPM works efficiently.

- Characterized in terms of a sparse Cholesky factorization
- Characterized in terms of a sparse chordal graph structure
- Useful to solve large-scale sparse POPs in practice

POP min. $f_0(\boldsymbol{x})$ s.t. $f_j(\boldsymbol{x}) \geq 0$ or $= 0$ $(j = 1, \ldots, m)$.

$\boldsymbol{H} f_0(\boldsymbol{x})$: the $n \times n$ Hessian mat. of $f_0(\boldsymbol{x})$,

$\boldsymbol{J}\boldsymbol{f}_*(\boldsymbol{x})$: the $m \times n$ Jacob. mat. of $\boldsymbol{f}_*(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^T$,

$\boldsymbol{R}$: the csp matrix, the $n \times n$ density pattern matrix of $\boldsymbol{I} + \boldsymbol{H} f_0(\boldsymbol{x}) + \boldsymbol{J}\boldsymbol{f}_*(\boldsymbol{x})^T \boldsymbol{J}\boldsymbol{f}_*(\boldsymbol{x})$ (no cancellation in '$+$').

$[\boldsymbol{J}\boldsymbol{f}_*(\boldsymbol{x})^T \boldsymbol{J}\boldsymbol{f}_*(\boldsymbol{x})]_{ij} \neq 0$ iff x_i and x_j are in a common constraint.

POP min. $f_0(\boldsymbol{x})$ s.t. $f_j(\boldsymbol{x}) \geq 0$ or $= 0$ $(j = 1, \ldots, m)$.

$\boldsymbol{H} f_0(\boldsymbol{x})$: the $n \times n$ Hessian mat. of $f_0(\boldsymbol{x})$,

$\boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})$: the $m \times n$ Jacob. mat. of $\boldsymbol{f}_*(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^T$,

$\boldsymbol{R}$: the csp matrix, the $n \times n$ density pattern matrix of $\boldsymbol{I} + \boldsymbol{H} f_0(\boldsymbol{x}) + \boldsymbol{J}\boldsymbol{f}_*(\boldsymbol{x})^T \boldsymbol{J}\boldsymbol{f}_*(\boldsymbol{x})$ (no cancellation in '$+$').

$[\boldsymbol{J}\boldsymbol{f}_*(\boldsymbol{x})^T \boldsymbol{J}\boldsymbol{f}_*(\boldsymbol{x})]_{ij} \neq 0$ iff x_i and x_j are in a common constraint.

Example: $f_0(\boldsymbol{x}) = \sum_{k=1}^{6} (-x_k^2)$
$f_j(\boldsymbol{x}) = 1 - x_j^2 - 2x_{j+1}^2 - x_6^2$ $(j = 1, 2, \ldots, 5)$.

$$\text{the csp matrix } \boldsymbol{R} = \begin{pmatrix} \star & \star & 0 & 0 & 0 & \star \\ \star & \star & \star & 0 & 0 & \star \\ 0 & \star & \star & \star & 0 & \star \\ 0 & 0 & \star & \star & \star & \star \\ 0 & 0 & 0 & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \end{pmatrix}$$

POP min. $f_0(\boldsymbol{x})$ s.t. $f_j(\boldsymbol{x}) \geq 0$ or $= 0$ $(j = 1, \ldots, m)$.

$\boldsymbol{H} f_0(\boldsymbol{x})$: the $n \times n$ Hessian mat. of $f_0(\boldsymbol{x})$,

$\boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})$: the $m \times n$ Jacob. mat. of $\boldsymbol{f}_*(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^T$,

$\boldsymbol{R}$: the csp matrix, the $n \times n$ density pattern matrix of $\boldsymbol{I} + \boldsymbol{H} f_0(\boldsymbol{x}) + \boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})^T \boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})$ (no cancellation in '$+$').

$[\boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})^T \boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})]_{ij} \neq 0$ iff x_i and x_j are in a common constraint.

POP min. $f_0(\boldsymbol{x})$ s.t. $f_j(\boldsymbol{x}) \geq 0$ or $= 0$ $(j = 1, \ldots, m)$.

$\boldsymbol{H} f_0(\boldsymbol{x})$: the $n \times n$ Hessian mat. of $f_0(\boldsymbol{x})$,

$\boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})$: the $m \times n$ Jacob. mat. of $\boldsymbol{f}_*(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^T$,

$\boldsymbol{R}$: the csp matrix, the $n \times n$ density pattern matrix of $\boldsymbol{I} + \boldsymbol{H} f_0(\boldsymbol{x}) + \boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})^T \boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})$ (no cancellation in '$+$').

$[\boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})^T \boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})]_{ij} \neq 0$ iff x_i and x_j are in a common constraint.

POP : c-sparse (correlatively sparse) $\Leftrightarrow$ The $n \times n$ csp matrix $\boldsymbol{R} = (R_{ij})$ allows a symbolic sparse Cholesky factorization (under a row & col. ordering like a symmetric min. deg. ordering).

$\boxed{\text{POP min. } f_0(\boldsymbol{x}) \text{ s.t. } f_j(\boldsymbol{x}) \geq 0 \text{ or } = 0 \ (j = 1, \ldots, m).}$

$\boldsymbol{H} f_0(\boldsymbol{x})$: the $n \times n$ Hessian mat. of $f_0(\boldsymbol{x})$,

$\boldsymbol{J}\boldsymbol{f}_*(\boldsymbol{x})$: the $m \times n$ Jacob. mat. of $\boldsymbol{f}_*(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^T$,

$\boldsymbol{R}$: the csp matrix, the $n \times n$ density pattern matrix of

$\boldsymbol{I} + \boldsymbol{H} f_0(\boldsymbol{x}) + \boldsymbol{J}\boldsymbol{f}_*(\boldsymbol{x})^T \boldsymbol{J}\boldsymbol{f}_*(\boldsymbol{x})$ (no cancellation in '$+$').

$[\boldsymbol{J}\boldsymbol{f}_*(\boldsymbol{x})^T \boldsymbol{J}\boldsymbol{f}_*(\boldsymbol{x})]_{ij} \neq 0$ iff x_i and x_j are in a common constraint.

POP min. $f_0(\boldsymbol{x})$ s.t. $f_j(\boldsymbol{x}) \geq 0$ or $= 0$ $(j = 1, \ldots, m)$.

$\boldsymbol{H} f_0(\boldsymbol{x})$: the $n \times n$ Hessian mat. of $f_0(\boldsymbol{x})$,

$\boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})$: the $m \times n$ Jacob. mat. of $\boldsymbol{f}_*(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^T$,

$\boldsymbol{R}$: the csp matrix, the $n \times n$ density pattern matrix of $\boldsymbol{I} + \boldsymbol{H} f_0(\boldsymbol{x}) + \boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})^T \boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})$ (no cancellation in '$+$').

$[\boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})^T \boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})]_{ij} \neq 0$ iff x_i and x_j are in a common constraint.

Example: $f_0(\boldsymbol{x}) = \sum_{k=1}^{6} (-x_k^2)$

$f_j(\boldsymbol{x}) = 1 - x_j^2 - 2x_{j+1}^2 - x_6^2$ $(j = 1, 2, \ldots, 5)$.

the csp matrix $\boldsymbol{R} = \begin{pmatrix} \star & \star & 0 & 0 & 0 & \star \\ \star & \star & \star & 0 & 0 & \star \\ 0 & \star & \star & \star & 0 & \star \\ 0 & 0 & \star & \star & \star & \star \\ 0 & 0 & 0 & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \end{pmatrix}$

tri-daig. + bordered

$\Downarrow$

No fill-in in Cholesky factorization

Contents

Sparse (SDP) relaxation = Lasserre (2001) + c-sparsity

POP min. $f_0(\boldsymbol{x})$ s.t. $f_j(\boldsymbol{x}) \geq 0$ or $= 0$ $(j = 1, \ldots, m)$, c-sparse.

$$\Downarrow$$

A sequence of c-sparse SDP relaxation problems depending on the relaxation order $r = 1, 2, \ldots$;

(a) Under a moderate assumption,
opt. sol. of SDP $\rightarrow$ opt sol. of POP as $r \rightarrow \infty$
(Lasserre 2006).

(b) $r = \lceil$"the max. deg. of poly. in POP"$/2\rceil + 0 \sim 3$ is usually large enough to attain opt sol. of POP in practice.

(c) Such an r is unknown in theory except $\exists$ special cases.

(d) The size of SDP increases as $r \rightarrow \infty$.

Example of Sparse SDP relaxation for POP with Inequalities

POP: min $\sum_{i=1}^{4}(-x_i^3)$ s.t. $-a_i \times x_i^2 - x_4^2 + 1 \geq 0 \; (i = 1, 2, 3)$.

Example of Sparse SDP relaxation for POP with Inequalities

POP: min $\sum_{i=1}^{4}(-x_i^3)$ s.t. $-a_i \times x_i^2 - x_4^2 + 1 \geq 0 \ (i = 1, 2, 3)$.

$\Updownarrow$ with the relaxation order $r = 2 \geq r_0 = \lceil 3/2 \rceil = 2$

poly.SDP:

min $\sum_{i=1}^{4}(-x_i^3)$

s.t. $(-a_i \times x_i^2 - x_4^2 + 1)(1, x_i, x_4)^T(1, x_i, x_4) \succeq \boldsymbol{O} \ \ i = 1, 2, 3,$

$(1, x_j, x_4, x_j^2, x_j x_4, x_4^2)^T(1, x_j, x_4, x_j^2, x_j x_4, x_4^2) \succeq \boldsymbol{O} \ j = 1, 2, 3.$

Example of Sparse SDP relaxation for POP with Inequalities

POP: min $\sum_{i=1}^{4}(-x_i^3)$ s.t. $-a_i \times x_i^2 - x_4^2 + 1 \geq 0$ $(i = 1, 2, 3)$.

$\Updownarrow$ with the relaxation order $r = 2 \geq r_0 = \lceil 3/2 \rceil = 2$

poly.SDP:

min $\sum_{i=1}^{4}(-x_i^3)$

s.t. $(-a_i \times x_i^2 - x_4^2 + 1)(1, x_i, x_4)^T(1, x_i, x_4) \succeq \boldsymbol{O}$ $\;i = 1, 2, 3,$

$(1, x_j, x_4, x_j^2, x_j x_4, x_4^2)^T(1, x_j, x_4, x_j^2, x_j x_4, x_4^2) \succeq \boldsymbol{O}$ $j = 1, 2, 3.$

Represent poly.SDP as

min $\sum_{\boldsymbol{\alpha} \in \mathcal{A}_0} g_0(\boldsymbol{\alpha})\boldsymbol{x}^{\boldsymbol{\alpha}}$ s.t. $\sum_{\boldsymbol{\alpha} \in \mathcal{A}_j} \boldsymbol{G}_j(\boldsymbol{\alpha})\boldsymbol{x}^{\boldsymbol{\alpha}} \succeq \boldsymbol{O}$ $j = 1, \ldots, 6,$

where $\mathcal{A}_j \subset \mathbb{Z}_+^4$ and $\boldsymbol{x}^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} x_3^{\alpha_3} x_4^{\alpha_4}$; $\boldsymbol{x}^{(1,2,1,0)} = x_1 x_2^2 x_3$.

Example of Sparse SDP relaxation for POP with Inequalities

POP: min $\sum_{i=1}^{4}(-x_i^3)$ s.t. $-a_i \times x_i^2 - x_4^2 + 1 \geq 0 \ (i = 1, 2, 3)$.

$\Updownarrow$ with the relaxation order $r = 2 \geq r_0 = \lceil 3/2 \rceil = 2$

poly.SDP:

min $\sum_{i=1}^{4}(-x_i^3)$

s.t. $(-a_i \times x_i^2 - x_4^2 + 1)(1, x_i, x_4)^T(1, x_i, x_4) \succeq \boldsymbol{O} \quad i = 1, 2, 3,$

$(1, x_j, x_4, x_j^2, x_j x_4, x_4^2)^T(1, x_j, x_4, x_j^2, x_j x_4, x_4^2) \succeq \boldsymbol{O} \ j = 1, 2, 3.$

Represent poly.SDP as

min $\sum_{\boldsymbol{\alpha} \in \mathcal{A}_0} g_0(\boldsymbol{\alpha}) \boldsymbol{x}^{\boldsymbol{\alpha}}$ s.t. $\sum_{\boldsymbol{\alpha} \in \mathcal{A}_j} \boldsymbol{G}_j(\boldsymbol{\alpha}) \boldsymbol{x}^{\boldsymbol{\alpha}} \succeq \boldsymbol{O} \ j = 1, \ldots, 6,$

where $\mathcal{A}_j \subset \mathbb{Z}_+^4$ and $\boldsymbol{x}^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} x_3^{\alpha_3} x_4^{\alpha_4}$; $\boldsymbol{x}^{(1,2,1,0)} = x_1 x_2^2 x_3$.

$\Downarrow$ Linearize by replacing each $\boldsymbol{x}^{\boldsymbol{\alpha}}$ by an indep. var. $y_{\boldsymbol{\alpha}}$; $\boldsymbol{x}^0$ by 1

SDP min $\displaystyle\sum_{\boldsymbol{\alpha} \in \mathcal{A}_0} g_0(\boldsymbol{\alpha}) y_{\boldsymbol{\alpha}}$ s.t. $\displaystyle\sum_{\boldsymbol{\alpha} \in \mathcal{A}_j} \boldsymbol{G}_j(\boldsymbol{\alpha}) y_{\boldsymbol{\alpha}} \succeq \boldsymbol{O} \ j = 1, \ldots, 6,$

which forms an SDP relaxation of POP.

Example of Sparse SDP relaxation for POPs with Equalities

POP: min $\sum_{i=1}^{4}(-x_i^3)$ s.t. $-a_i \times x_i^2 - x_4^2 + 1 = 0 \; i = 1, 2, 3$.

Example of Sparse SDP relaxation for POPs with Equalities

POP: min $\sum_{i=1}^{4}(-x_i^3)$ s.t. $-a_i \times x_i^2 - x_4^2 + 1 = 0 \; i = 1,2,3.$

$\Updownarrow$ with the relaxation order $r = 2 \geq r_0 = \lceil 3/2 \rceil = 2$

poly.SDP:

min $\sum_{i=1}^{4}(-x_i^3)$

s.t. $(-a_i \times x_i^2 - x_4^2 + 1)(1, x_i, x_4, x_i^2, x_i x_4, x_4^2)^T = \mathbf{0} \;\; i = 1,2,3,$

$(1, x_j, x_4, x_j^2, x_j x_4, x_4^2)^T(1, x_j, x_4, x_j^2, x_j x_4, x_4^2) \succeq \boldsymbol{O} \; j = 1,2,3.$

$\Downarrow$ Represent poly.SDP as

min $\sum_{\boldsymbol{\alpha}\in\mathcal{A}_0} g_0(\boldsymbol{\alpha})\boldsymbol{x}^{\boldsymbol{\alpha}}$ s.t. $\sum_{\boldsymbol{\alpha}\in\mathcal{A}_i} \boldsymbol{g}_i \boldsymbol{x}^{\boldsymbol{\alpha}} = \mathbf{0}$

$\sum_{\boldsymbol{\alpha}\in\mathcal{A}_j} \boldsymbol{G}_j(\boldsymbol{\alpha})\boldsymbol{x}^{\boldsymbol{\alpha}} \succeq \boldsymbol{O} \; j = 1,2,3$

$\Downarrow$ Linearize by replacing each $\boldsymbol{x}^{\boldsymbol{\alpha}}$ by an indep. var. $y_{\boldsymbol{\alpha}}$; $\boldsymbol{x}^0$ by 1

min $\sum_{\boldsymbol{\alpha}\in\mathcal{A}_0} g_0(\boldsymbol{\alpha})y_{\boldsymbol{\alpha}}$ s.t. $\sum_{\boldsymbol{\alpha}\in\mathcal{A}_i} \boldsymbol{g}_i y_{\boldsymbol{\alpha}} = \mathbf{0}$

$\sum_{\boldsymbol{\alpha}\in\mathcal{A}_j} \boldsymbol{G}_j(\boldsymbol{\alpha})\boldsymbol{y}_{\boldsymbol{\alpha}} \succeq \boldsymbol{O} \; j = 1,2,3$

Example of Sparse SDP relaxation for POPs with Equalities

POP: min $\sum_{i=1}^{4}(-x_i^3)$ s.t. $-a_i \times x_i^2 - x_4^2 + 1 = 0 \; i = 1, 2, 3.$

$\Updownarrow$ with the relaxation order $r = 2 \geq r_0 = \lceil 3/2 \rceil = 2$

poly.SDP:

min $\sum_{i=1}^{4}(-x_i^3)$

s.t. $(-a_i \times x_i^2 - x_4^2 + 1)(1, x_i, x_4, x_i^2, x_i x_4, x_4^2)^T = \mathbf{0} \;\; i = 1, 2, 3,$

$(1, x_j, x_4, x_j^2, x_j x_4, x_4^2)^T (1, x_j, x_4, x_j^2, x_j x_4, x_4^2) \succeq \boldsymbol{O} \; j = 1, 2, 3.$

$\Downarrow$ Represent poly.SDP as

min $\sum_{\boldsymbol{\alpha} \in \mathcal{A}_0} g_0(\boldsymbol{\alpha}) \boldsymbol{x}^{\boldsymbol{\alpha}}$ s.t. $\sum_{\boldsymbol{\alpha} \in \mathcal{A}_i} \boldsymbol{g}_i \boldsymbol{x}^{\boldsymbol{\alpha}} = \mathbf{0}$

$\sum_{\boldsymbol{\alpha} \in \mathcal{A}_j} \boldsymbol{G}_j(\boldsymbol{\alpha}) \boldsymbol{x}^{\boldsymbol{\alpha}} \succeq \boldsymbol{O} \; j = 1, 2, 3$

$\Downarrow$ Linearize by replacing each $\boldsymbol{x}^{\boldsymbol{\alpha}}$ by an indep. var. $y_{\boldsymbol{\alpha}}$; $\boldsymbol{x}^0$ by 1

min $\sum_{\boldsymbol{\alpha} \in \mathcal{A}_0} g_0(\boldsymbol{\alpha}) y_{\boldsymbol{\alpha}}$ s.t. $\sum_{\boldsymbol{\alpha} \in \mathcal{A}_i} \boldsymbol{g}_i y_{\boldsymbol{\alpha}} = \mathbf{0}$

$\sum_{\boldsymbol{\alpha} \in \mathcal{A}_j} \boldsymbol{G}_j(\boldsymbol{\alpha}) \boldsymbol{y}_{\boldsymbol{\alpha}} \succeq \boldsymbol{O} \; j = 1, 2, 3$

Equalities in dual SDP $\Leftrightarrow$ Free variables in primal SDP $\Rightarrow$ next

Contents

Contents

- How to handle free variables is an important issue in primal-dual interior-point methods for SDPs.
- Some methods have been developed;
  free $z = z_+ - z_-$, $z_+, \ z_- \geq 0$, using a second order cone.

A new method $\Rightarrow$

Primal SDP having free vector variable $\boldsymbol{z}$

$$\mathcal{P}: \quad \begin{array}{ll} \text{min} & \boldsymbol{d}^T\boldsymbol{z} + \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} & \boldsymbol{D}\boldsymbol{z} + \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \succeq \mathbf{0}, \end{array} \qquad \begin{array}{l} \boldsymbol{D} : m \times k,\ \text{rank}\ \boldsymbol{D} = k, \\ \boldsymbol{A} : m \times n,\ \text{where}\ m \geq k \end{array}$$

Dual SDP having equality constraints

$$\mathcal{D}: \quad \text{max} \quad \boldsymbol{b}^T\boldsymbol{y} \quad \text{s.t.} \quad \boldsymbol{D}^T\boldsymbol{y} = \boldsymbol{d},\ \boldsymbol{A}^T\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c},\ \boldsymbol{s} \succeq \mathbf{0}.$$

Primal SDP having free vector variable $\boldsymbol{z}$

$$\mathcal{P}: \quad \begin{array}{ll} \min & \boldsymbol{d}^T\boldsymbol{z} + \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} & \boldsymbol{D}\boldsymbol{z} + \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \succeq \boldsymbol{0}, \end{array} \qquad \begin{array}{l} \boldsymbol{D} : m \times k,\ \text{rank } \boldsymbol{D} = k, \\ \boldsymbol{A} : m \times n,\ \text{where } m \geq k \end{array}$$

Dual SDP having equality constraints

$$\mathcal{D}: \quad \max \quad \boldsymbol{b}^T\boldsymbol{y} \quad \text{s.t.} \quad \boldsymbol{D}^T\boldsymbol{y} = \boldsymbol{d},\ \boldsymbol{A}^T\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c},\ \boldsymbol{s} \succeq \boldsymbol{0}.$$

- Primal approach: Eliminate free variable $\boldsymbol{z}$ by pivoting $\Rightarrow$

$$\widehat{\mathcal{P}}: \quad \begin{array}{ll} \min & \hat{\boldsymbol{c}}^T\boldsymbol{x} + \hat{\gamma} \\ \text{s.t.} & \widehat{\boldsymbol{A}}_2\boldsymbol{x} = \hat{\boldsymbol{b}}_2,\ \boldsymbol{x} \succeq \boldsymbol{0}, \end{array} \qquad \widehat{\boldsymbol{A}}_2 : (m-k) \times n.$$

- Dual : Solve $\boldsymbol{D}^T\boldsymbol{y} = \boldsymbol{d}$ in $\boldsymbol{y}_1$, $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2) \in \mathbb{R}^{k+(m-k)}$.

$$\widehat{\mathcal{D}}: \quad \max \quad \hat{\boldsymbol{b}}_2^T\boldsymbol{y}_2 + \hat{\gamma} \quad \text{s.t.} \quad \widehat{\boldsymbol{A}}_2^T\boldsymbol{y}_2 + \boldsymbol{s} = \hat{\boldsymbol{c}},\ \boldsymbol{s} \succeq \boldsymbol{0}.$$

- The size gets smaller, but $\widehat{\boldsymbol{A}}_2$ could get denser than $\boldsymbol{A}$.
- Numerical stability in pivoting or solving $\boldsymbol{D}^T\boldsymbol{y} = \boldsymbol{d}$ in $\boldsymbol{y}_1$.

Primal SDP having free vector variable $\boldsymbol{z}$

$$\mathcal{P}: \quad \begin{array}{ll} \text{min} & \boldsymbol{d}^T\boldsymbol{z} + \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} & \boldsymbol{D}\boldsymbol{z} + \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \succeq \mathbf{0}, \end{array} \qquad \begin{array}{l} \boldsymbol{D} : m \times k,\ \text{rank}\ \boldsymbol{D} = k, \\ \boldsymbol{A} : m \times n,\ \text{where}\ m \geq k \end{array}$$

Primal SDP having free vector variable $\boldsymbol{z}$

$$\mathcal{P}: \quad \begin{array}{ll} \text{min} & \boldsymbol{d}^T\boldsymbol{z} + \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} & \boldsymbol{D}\boldsymbol{z} + \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \ \boldsymbol{x} \succeq \boldsymbol{0}, \end{array} \qquad \begin{array}{l} \boldsymbol{D} : m \times k, \ \text{rank } \boldsymbol{D} = k, \\ \boldsymbol{A} : m \times n, \ \text{where } m \geq k \end{array}$$

A stable sparse LU factorization to $\boldsymbol{D}$ for simplisity,

$$\boldsymbol{P}\boldsymbol{D}\boldsymbol{Q} \ = \ \boldsymbol{L}\boldsymbol{U} \quad \text{or} \quad \boldsymbol{D} = \boldsymbol{P}^T\boldsymbol{L}\boldsymbol{U}\boldsymbol{Q}^T \qquad = \boldsymbol{L}\boldsymbol{U}$$

$$k, \qquad \boldsymbol{U} \ : \ k \times k \text{ upper triangular},$$

$$\boldsymbol{L} \ = \ \begin{array}{cc} \begin{pmatrix} \boldsymbol{L}_1 \\ \boldsymbol{L}_2 \end{pmatrix} & \begin{array}{c} k \\ m-k \end{array} \end{array}, \ \boldsymbol{L}_1 \ : \ \text{lower triangular},$$

$$\boldsymbol{P} \ : \ \text{an } m \times m \text{ permutation matrix}, \qquad = \boldsymbol{I}$$

$$\boldsymbol{Q} \ : \ \text{a } k \times k \text{ permutation matrix}, \qquad = \boldsymbol{I}$$

Primal SDP having free vector variable $\boldsymbol{z}$

$$\mathcal{P}: \quad \begin{array}{ll} \text{min} & \boldsymbol{d}^T\boldsymbol{z} + \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} & \boldsymbol{D}\boldsymbol{z} + \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \succeq \mathbf{0}, \end{array} \qquad \begin{array}{l} \boldsymbol{D} : m \times k,\ \text{rank}\ \boldsymbol{D} = k, \\ \boldsymbol{A} : m \times n,\ \text{where}\ m \geq k \end{array}$$

A stable sparse LU factorization to $\boldsymbol{D}$

$$\boldsymbol{D} = \boldsymbol{L}\boldsymbol{U}$$

$$\begin{array}{ll} k, & \boldsymbol{U} : k \times k \text{ upper triangular,} \end{array}$$

$$\boldsymbol{L} = \begin{pmatrix} \boldsymbol{L}_1 \\ \boldsymbol{L}_2 \end{pmatrix} \begin{array}{c} k \\ m-k \end{array},\ \boldsymbol{L}_1 : \text{ lower triangular,}$$

Primal SDP having free vector variable $\boldsymbol{z}$

$$\mathcal{P}: \quad \begin{array}{ll} \text{min} & \boldsymbol{d}^T\boldsymbol{z} + \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} & \boldsymbol{D}\boldsymbol{z} + \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \succeq \boldsymbol{0}, \end{array} \qquad \begin{array}{l} \boldsymbol{D} : m \times k,\ \text{rank}\ \boldsymbol{D} = k, \\ \boldsymbol{A} : m \times n,\ \text{where}\ m \geq k \end{array}$$

A stable sparse LU factorization to $\boldsymbol{D}$

$$\boldsymbol{D} = \boldsymbol{L}\boldsymbol{U}$$

$$k, \qquad \boldsymbol{U} : k \times k \text{ upper triangular},$$

$$\boldsymbol{L} = \begin{pmatrix} \boldsymbol{L}_1 \\ \boldsymbol{L}_2 \end{pmatrix} \begin{array}{c} k \\ m-k \end{array}, \quad \boldsymbol{L}_1 : \text{ lower triangular},$$

$$\widehat{\mathcal{P}}: \quad \begin{array}{ll} \text{min} & \hat{\boldsymbol{c}}^T\boldsymbol{x} + \hat{\gamma} \\ \text{s.t.} & \widehat{\boldsymbol{A}}_2\boldsymbol{x} = \hat{\boldsymbol{b}}_2,\ \boldsymbol{x} \succeq \boldsymbol{0},\ \boldsymbol{z} = \boldsymbol{U}^{-1}(\hat{\boldsymbol{b}}_1 - \widehat{\boldsymbol{A}}_1\boldsymbol{x}). \end{array} \qquad \widehat{\boldsymbol{A}}_2 : (m-k) \times n,\ \widehat{\boldsymbol{A}}_1 : k \times n$$

$$\hat{\boldsymbol{c}} = \boldsymbol{c} - \widehat{\boldsymbol{A}}_1^T\boldsymbol{U}^{-T}\boldsymbol{d},\ \hat{\gamma} = \hat{\boldsymbol{b}}_1^T\boldsymbol{U}^{-T}\boldsymbol{d},$$

$$\begin{pmatrix} \widehat{\boldsymbol{A}}_1 \\ \widehat{\boldsymbol{A}}_2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{L}_1 & \boldsymbol{O} \\ \boldsymbol{L}_2 & \boldsymbol{I} \end{pmatrix}^{-1} \boldsymbol{A},\ \begin{pmatrix} \hat{\boldsymbol{b}}_1 \\ \hat{\boldsymbol{b}}_2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{L}_1 & \boldsymbol{O} \\ \boldsymbol{L}_2 & \boldsymbol{I} \end{pmatrix}^{-1} \boldsymbol{b},$$

Primal SDP having free vector variable $\boldsymbol{z}$

$$\mathcal{P}: \quad \begin{array}{ll} \text{min} & \boldsymbol{d}^T\boldsymbol{z} + \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} & \boldsymbol{D}\boldsymbol{z} + \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \ \boldsymbol{x} \succeq \boldsymbol{0}, \end{array} \qquad \begin{array}{l} \boldsymbol{D} : m \times k, \ \text{rank } \boldsymbol{D} = k, \\ \boldsymbol{A} : m \times n, \ \text{where } m \geq k \end{array}$$

A stable sparse LU factorization to $\boldsymbol{D}$

$$\boldsymbol{D} = \boldsymbol{L}\boldsymbol{U}$$

$$k, \qquad \boldsymbol{U} : k \times k \text{ upper triangular,}$$

$$\boldsymbol{L} = \begin{array}{c} \begin{pmatrix} \boldsymbol{L}_1 \\ \boldsymbol{L}_2 \end{pmatrix} \end{array} \begin{array}{c} k \\ m-k \end{array}, \ \boldsymbol{L}_1 : \text{ lower triangular,}$$

$$\boxed{\widehat{\mathcal{P}}: \quad \begin{array}{ll} \text{min} & \hat{\boldsymbol{c}}^T\boldsymbol{x} + \hat{\gamma} \\ \text{s.t.} & \widehat{\boldsymbol{A}}_2\boldsymbol{x} = \hat{\boldsymbol{b}}_2, \ \boldsymbol{x} \succeq \boldsymbol{0}, \end{array} \quad \begin{array}{l} \widehat{\boldsymbol{A}}_2 : (m-k) \times n, \ \widehat{\boldsymbol{A}}_1 : k \times n \\ \boldsymbol{z} = \boldsymbol{U}^{-1}(\hat{\boldsymbol{b}}_1 - \widehat{\boldsymbol{A}}_1\boldsymbol{x}). \end{array}}$$

Primal SDP having free vector variable $\boldsymbol{z}$

$$\mathcal{P}: \quad \begin{array}{ll} \text{min} & \boldsymbol{d}^T\boldsymbol{z} + \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} & \boldsymbol{D}\boldsymbol{z} + \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \succeq \boldsymbol{0}, \end{array} \qquad \begin{array}{l} \boldsymbol{D} : m \times k,\ \text{rank}\ \boldsymbol{D} = k, \\ \boldsymbol{A} : m \times n,\ \text{where}\ m \geq k \end{array}$$

A stable sparse LU factorization to $\boldsymbol{D}$

$$\begin{array}{rcl} \boldsymbol{D} & = & \boldsymbol{L}\boldsymbol{U} \end{array}$$

$$k, \qquad \boldsymbol{U} : k \times k \text{ upper triangular,}$$

$$\boldsymbol{L} = \begin{pmatrix} \boldsymbol{L}_1 \\ \boldsymbol{L}_2 \end{pmatrix} \begin{array}{c} k \\ m-k \end{array}, \quad \boldsymbol{L}_1 : \text{ lower triangular,}$$

$$\widehat{\mathcal{P}}: \quad \begin{array}{ll} \text{min} & \hat{\boldsymbol{c}}^T\boldsymbol{x} + \hat{\gamma} \\ \text{s.t.} & \widehat{\boldsymbol{A}}_2\boldsymbol{x} = \hat{\boldsymbol{b}}_2,\ \boldsymbol{x} \succeq \boldsymbol{0},\ \boldsymbol{z} = \boldsymbol{U}^{-1}(\hat{\boldsymbol{b}}_1 - \widehat{\boldsymbol{A}}_1\boldsymbol{x}). \end{array} \qquad \widehat{\boldsymbol{A}}_2 : (m-k) \times n,\ \widehat{\boldsymbol{A}}_1 : k \times n$$

- k is larger $\Rightarrow$ smaller size
- $\boldsymbol{LU}$ factorization is well-conditioned $\Rightarrow$ higher accuracy
- $\boldsymbol{LU}$ factorization (or $\widehat{\boldsymbol{A}}_2$) is sparser $\Rightarrow$ more efficient
- can be applied to LP and SOCP

Contents

- All the methods described in Sections 3, 4 and 5 are applied in this section.
- Ongoing joint work with Kim and Waki

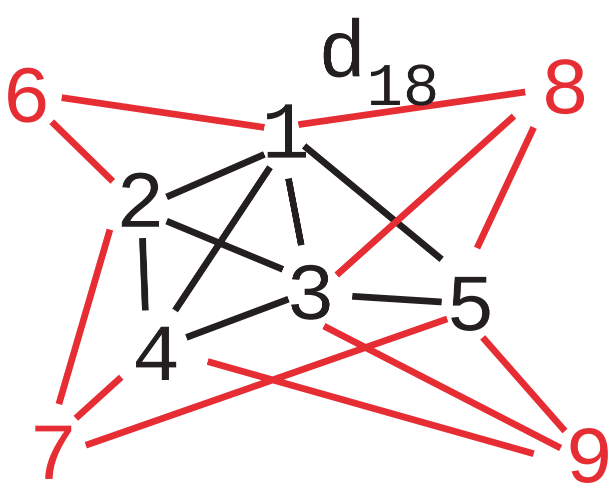
Sensor network localization problem: Let $s = 2$ or 3.

$$\begin{aligned} \boldsymbol{x}^p \in \mathbb{R}^s \quad &: \quad \text{unknown location of sensors } (p = 1, 2, \ldots, m), \\ \boldsymbol{x}^r = \boldsymbol{a}^r \in \mathbb{R}^s \quad &: \quad \text{known location of anchors } (r = m+1, \ldots, n), \\ d_{pq} \quad &= \quad \|\boldsymbol{x}^p - \boldsymbol{x}^q\| + \epsilon_{pq} \text{ — given for } (p, q) \in \mathcal{N}, \\ \mathcal{N} \quad &= \quad \{(p, q) : \|\boldsymbol{x}^p - \boldsymbol{x}^q\| \le \rho = \text{a given radio range}\} \end{aligned}$$

Here ϵ_{pq} denotes a noise.

$m = 5, \ n = 9$.
$1, \ldots, 5$: sensors
$6, 7, 8, 9$: anchors

Anchors' positions are fixed.
A distance is given for $\forall$ edge.
Compute locations of sensors.



- SDP relaxations Biswas et al. ’06, Nie ’06, ... for $s = 2$.
- An SOCP relaxation Tseng ’07 for $s = 2$.
- .....

$\Rightarrow$ Exploiting correlative sparsity in our new SDP relaxation

Basic idea of Sparse SDP relaxation

QOP: min $\sum_{pq}(v_{pq} - d_{pq})^2 \quad \equiv 0$

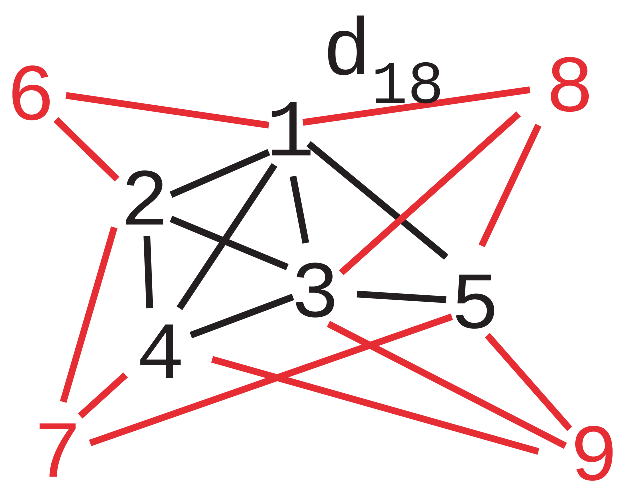
s.t $v_{pq}^2 = \|\boldsymbol{x}^p - \boldsymbol{x}^q\|^2 \ (p,q) \in \mathcal{N}, \ \boldsymbol{x}^r = \boldsymbol{a}^r \ (r > m),$

$0 \leq (1-\gamma)d_{pq} \leq v_{pq} \leq (1+\delta)d_{pq} \ (p,q) \in \mathcal{N}.$

Here $0 \leq \gamma \leq 1$, $0 \leq \delta$; $\quad \gamma = \delta = 0$ or $d_{pq} = v_{pq}$ if $\epsilon_{pq} \equiv 0$

Anchors' positions are fixed.
A distance is given for $\forall$ edge.
Compute locations of sensors.

$m = 5, \ n = 9$.
$1, \ldots, 5$: sensors
$6, 7, 8, 9$: anchors



Basic idea of Sparse SDP relaxation

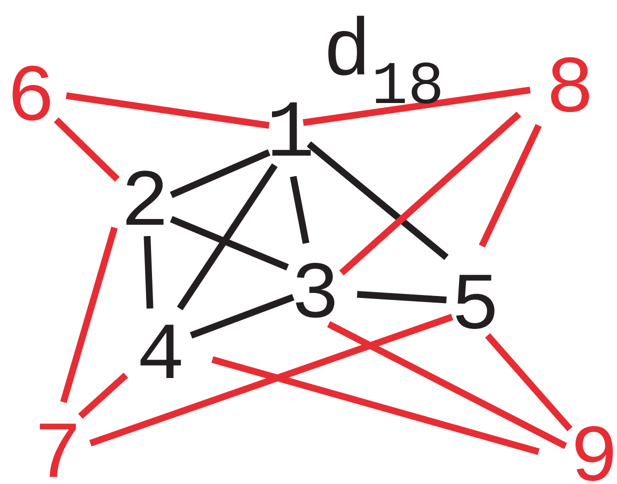
QOP: min $\sum_{pq}(v_{pq} - d_{pq})^2 \qquad \equiv 0$

s.t $v_{pq}^2 = \|\boldsymbol{x}^p - \boldsymbol{x}^q\|^2 \ (p,q) \in \mathcal{N}, \ \boldsymbol{x}^r = \boldsymbol{a}^r \ (r > m),$

$0 \leq (1-\gamma)d_{pq} \leq v_{pq} \leq (1+\delta)d_{pq} \ (p,q) \in \mathcal{N}.$

Here $0 \leq \gamma \leq 1$, $0 \leq \delta$; $\qquad \gamma = \delta = 0$ or $d_{pq} = v_{pq}$ if $\epsilon_{pq} \equiv 0$

$m = 5, \ n = 9.$
$1, \ldots, 5$: sensors
$6, 7, 8, 9$: anchors



Anchors' positions are fixed.
A distance is given for $\forall$ edge.
Compute locations of sensors.

- Remove some edges to reduce the size.
- Keep red edges in this example.
- Remove black edges as long as deg. of $\forall$ node $\geq \delta$; $\delta = 4$ or 5.

Basic idea of $\boxed{\text{Sparse SDP relaxation}}$

QOP: min $\sum_{pq}(v_{pq} - d_{pq})^2 \qquad \equiv 0$

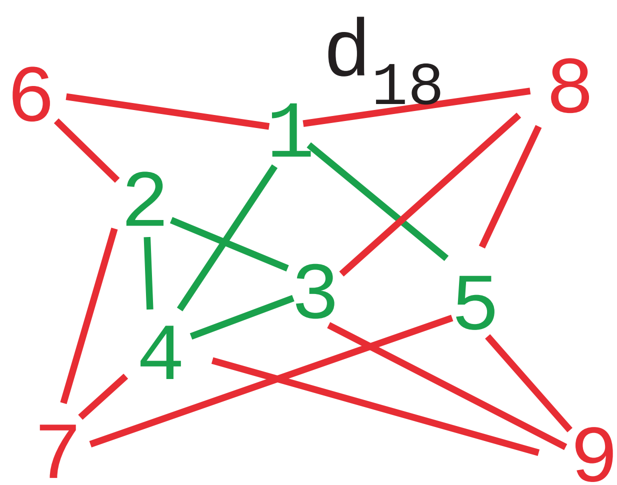
s.t $v_{pq}^2 = \|\boldsymbol{x}^p - \boldsymbol{x}^q\|^2 \ (p,q) \in \mathcal{N}, \ \boldsymbol{x}^r = \boldsymbol{a}^r \ (r > m),$

$0 \leq (1-\gamma)d_{pq} \leq v_{pq} \leq (1+\delta)d_{pq} \ (p,q) \in \mathcal{N}.$

Here $0 \leq \gamma \leq 1$, $0 \leq \delta$; $\qquad \gamma = \delta = 0$ or $d_{pq} = v_{pq}$ if $\epsilon_{pq} \equiv 0$

Anchors' positions are fixed.
A distance is given for $\forall$ edge.
Compute locations of sensors.

$m = 5, \ n = 9$.
$1, \ldots, 5$: sensors
$6, 7, 8, 9$: anchors



- Remove some edges to reduce the size.
- Keep red edges in this example.
- Remove black edges as long as deg. of $\forall$ node $\geq \delta$; $\delta = 4$ or 5.

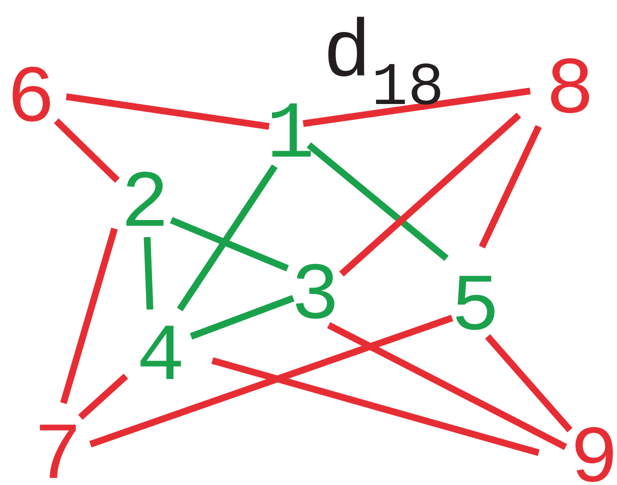
Basic idea of $\boxed{\text{Sparse SDP relaxation}}$

$$\text{QOP: min } \sum_{pq}(v_{pq} - d_{pq})^2 \qquad \equiv 0$$
$$\text{s.t } v_{pq}^2 = \|\boldsymbol{x}^p - \boldsymbol{x}^q\|^2 \ (p,q) \in \mathcal{N}, \ \boldsymbol{x}^r = \boldsymbol{a}^r \ (r > m),$$
$$0 \leq (1-\gamma)d_{pq} \leq v_{pq} \leq (1+\delta)d_{pq} \ (p,q) \in \mathcal{N}.$$

Here $0 \leq \gamma \leq 1$, $0 \leq \delta$; $\quad \gamma = \delta = 0$ or $d_{pq} = v_{pq}$ if $\epsilon_{pq} \equiv 0$

Anchors' positions are fixed.
A distance is given for $\forall$ edge.
Compute locations of sensors.

$m = 5, \ n = 9$.
$1, \ldots, 5$: sensors
$6, 7, 8, 9$: anchors



- Remove some edges to reduce the size.
- Keep red edges in this example.
- Remove black edges as long as deg. of $\forall$ node $\geq \delta$; $\delta = 4$ or 5.
- Use the red & green edges for $\mathcal{N}$ $\Rightarrow$ c-sparsity in QOP
- How we select the red & green edges for $\mathcal{N}$ is essential.

Preliminary numerical results

- Software — SparsePOP† + SeDuMi + Nonlinear LS meth.
- CPU 2.66 GHz Dual-Core Intel Xeon, memory 4 GB

† : Waki, S. Kim, M. Kojima and M. Muramatsu
"SparsePOP : a Sparse Semidefinite Programming Relaxation of Polynomial Optimization Problems"
March 2005. Revised August 2007.

Nonlinear LS method

to refine solutions computed by SparsePOP

$\Uparrow$

MATLAB function lsqnonlin

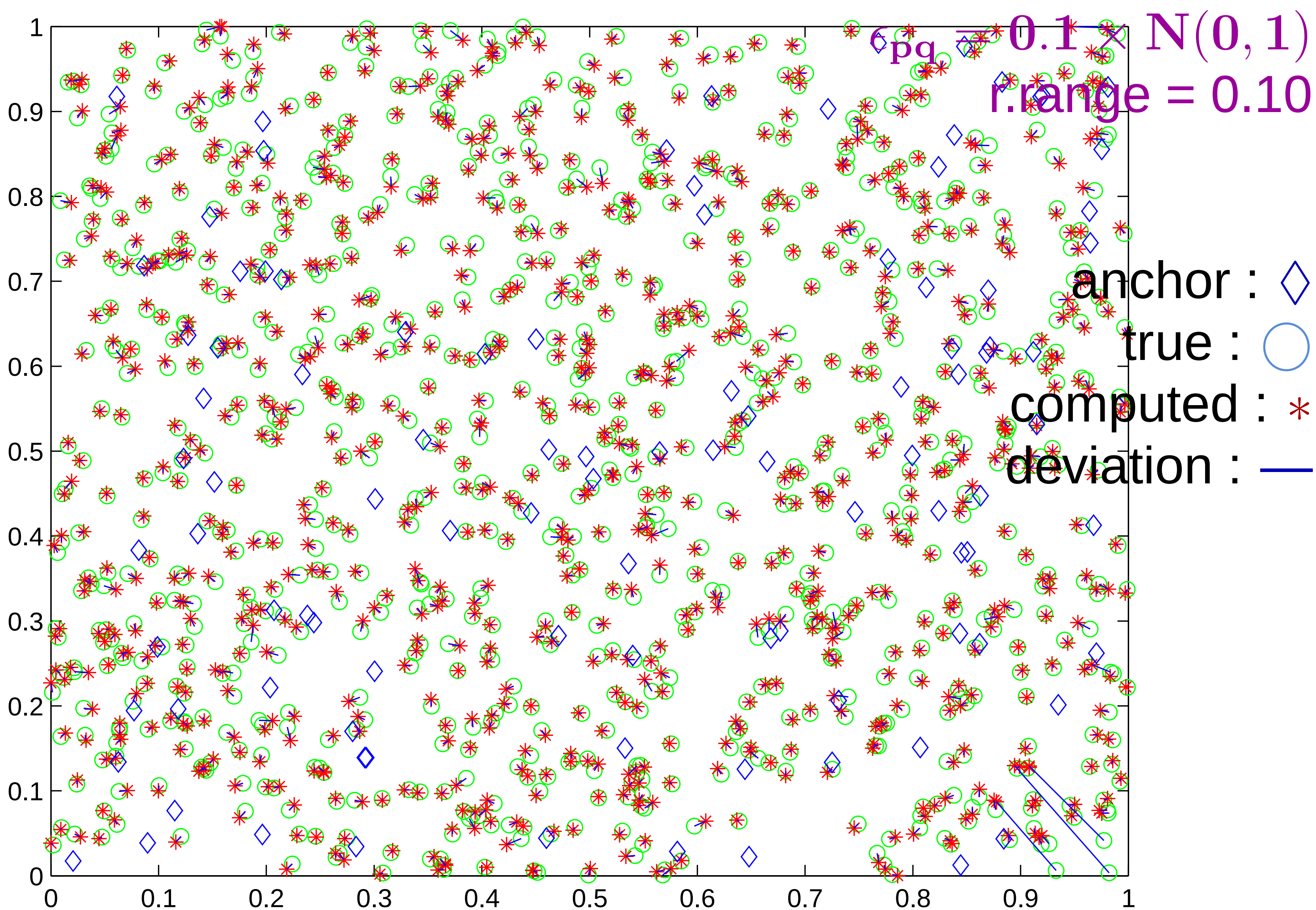
900 sensors and 100 anchors randomly distributed on $[0,1] \times [0,1]$

radio	No Noise: $\epsilon_{pq} = 0$		Noisy: $\epsilon_{pq} \sim 0.1 \times N(0,1)$	
range	rmsd	cpu	rmsd	cpu
0.1	3.1e-09	25.4	3.3e-04	204.9
0.2	1.1e-09	8.5	4.5e-04	173.8

rmsd $= \frac{1}{m}\left(\sum_{p=1}^{m} \|\boldsymbol{x}^p - \boldsymbol{a}^p\|^2\right)^{1/2}$ (root mean square distance)

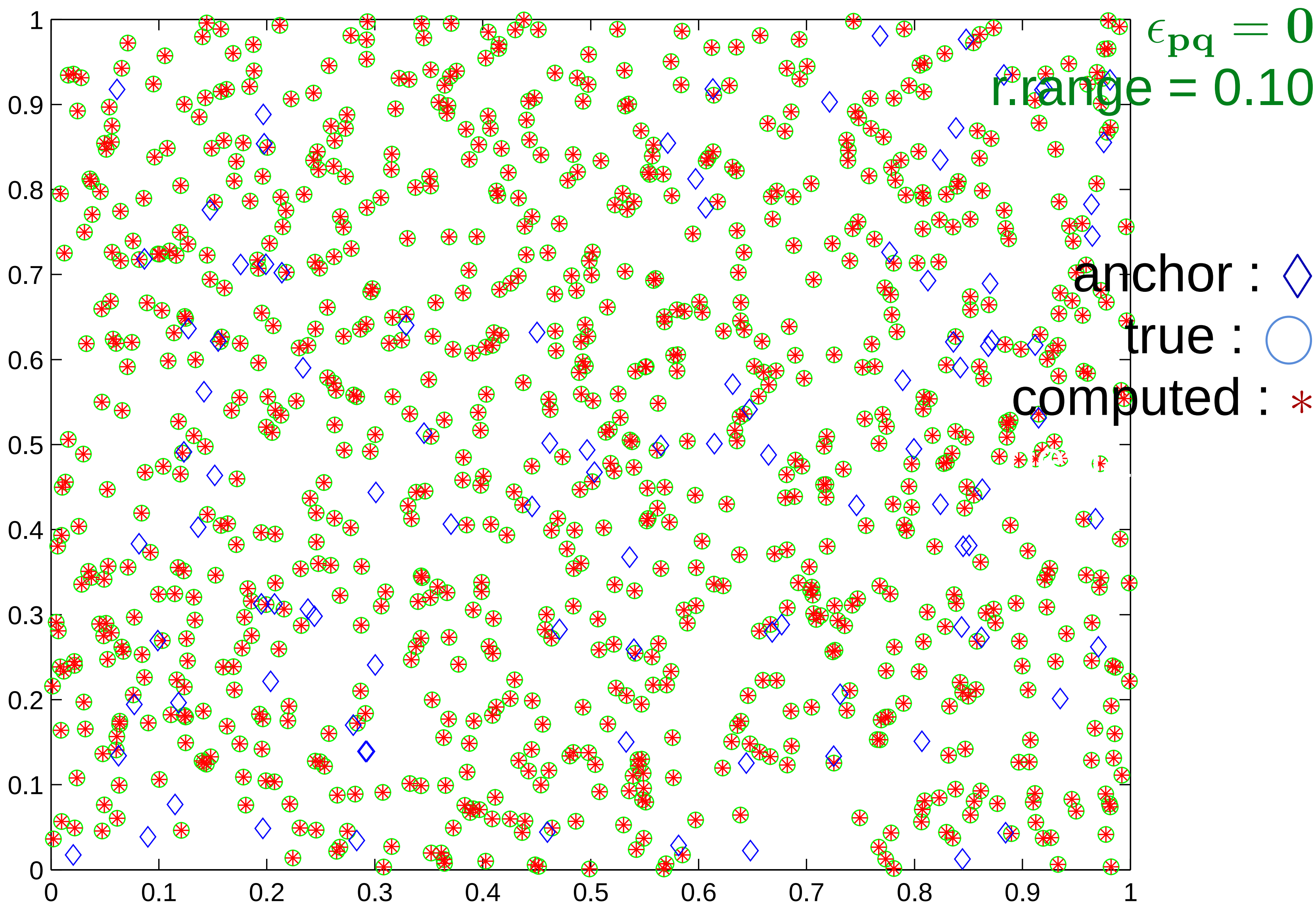
cpu = SeDuMi cpu time in second ($\not\supset$ conversion time)

900 sensors and 100 anchors on $[0,1] \times [0,1]$



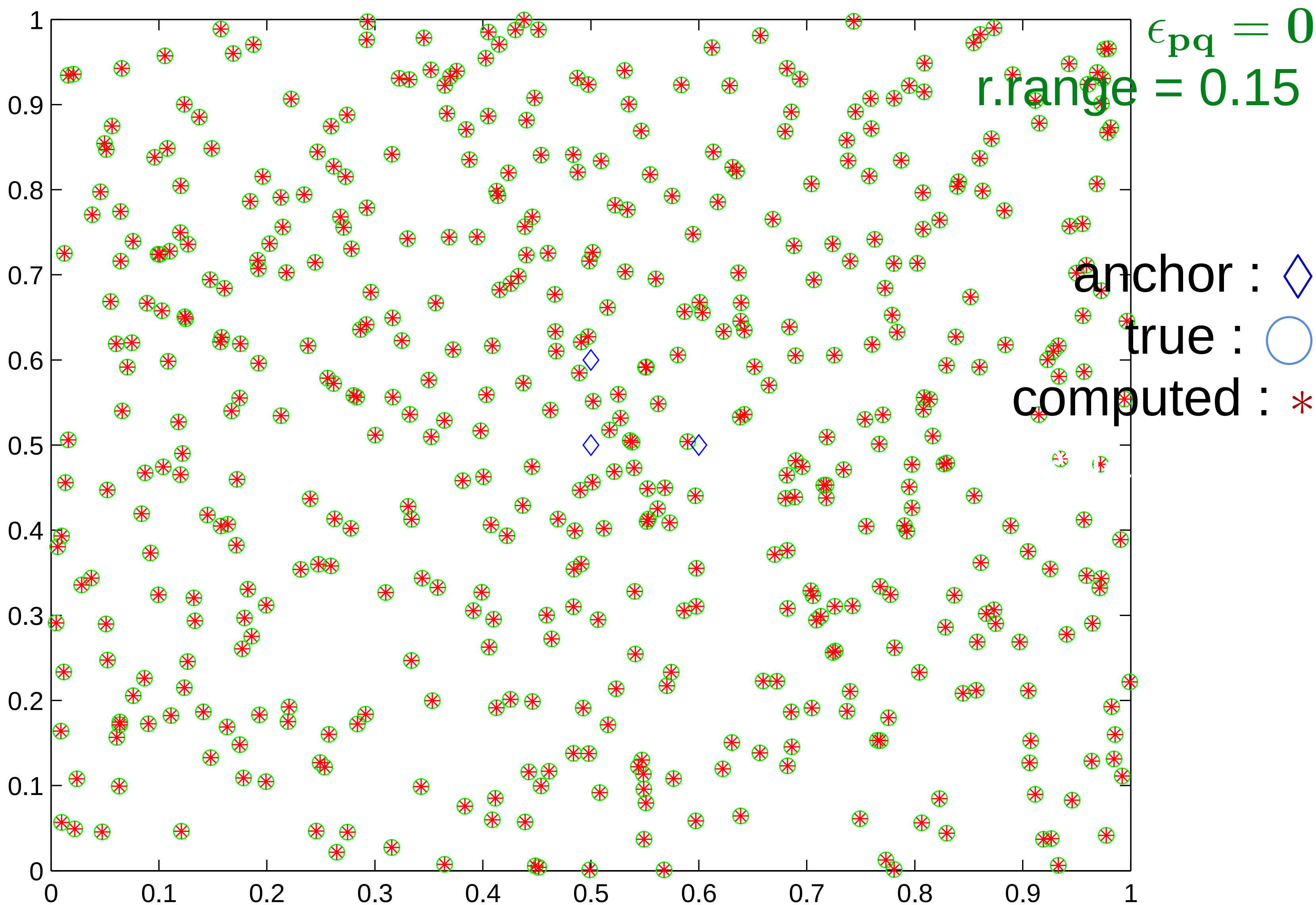
204.9 sec., rmsd $= \frac{1}{m}\left(\sum_{p=1}^{m} \|\boldsymbol{x}^p - \boldsymbol{a}^p\|^2\right)^{1/2}$ = 3.3e-04

900 sensors and 100 anchors on $[0,1] \times [0,1]$



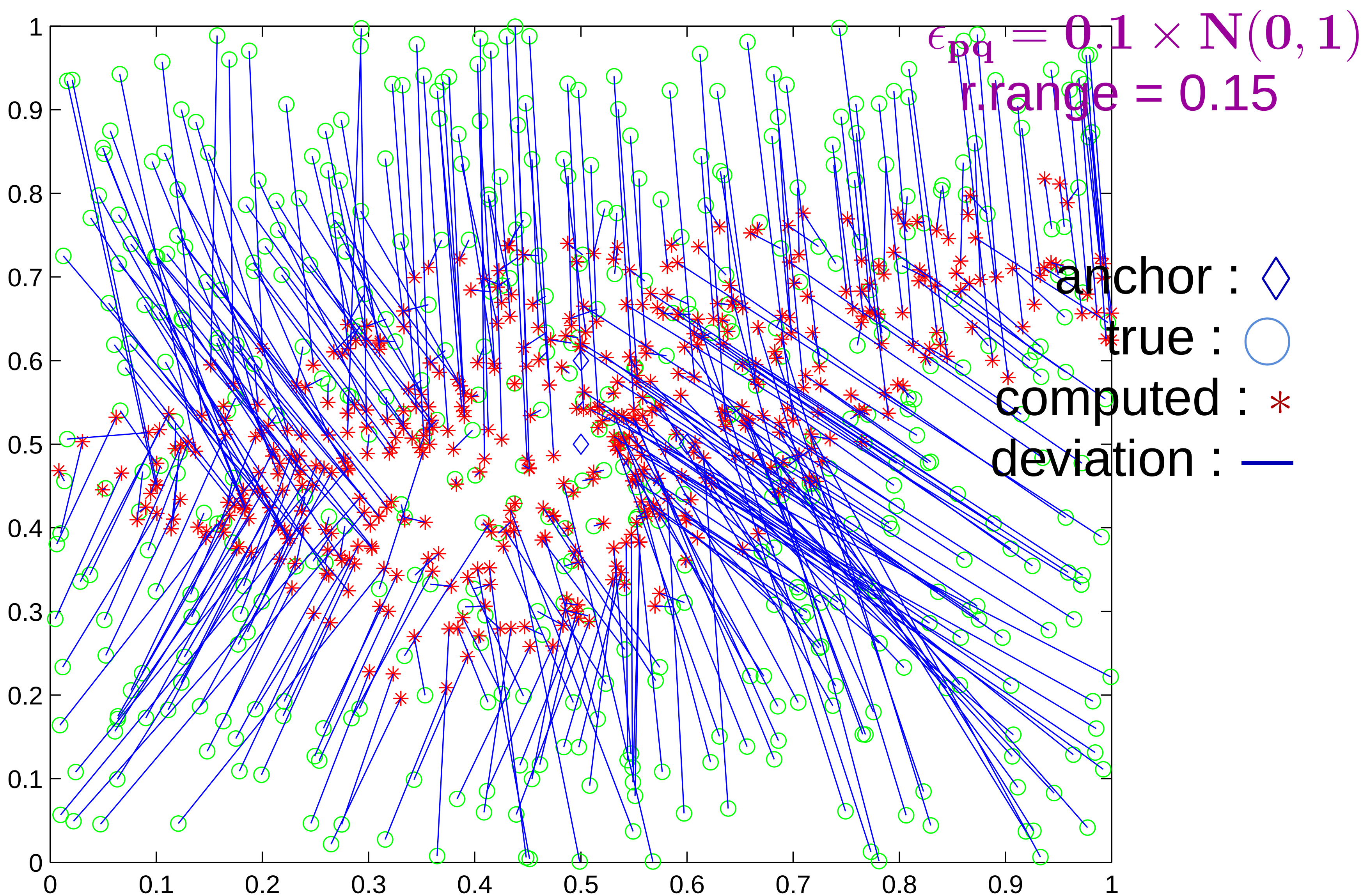
25.4 sec., rmsd $= \frac{1}{m}\left(\sum_{p=1}^{m} \|\boldsymbol{x}^p - \boldsymbol{a}^p\|^2\right)^{1/2}$ = 3.1e-09

500 sensors and 3 anchors at $(\mathbf{0.5, 0.5})$, $(\mathbf{0.6, 0.5})$, $(\mathbf{0.5, 0.6})$



320.0 sec., rmsd $= \frac{1}{m}\left(\sum_{p=1}^{m}\|\boldsymbol{x}^p - \boldsymbol{a}^p\|^2\right)^{1/2}$ = 5.7e-09

500 sensors and 3 anchors at $(0.5, 0.5),\ (0.6, 0.5),\ (0.5, 0.6)$



1324.5 sec., rmsd $= \dfrac{1}{m}\left(\sum_{p=1}^{m} \|\boldsymbol{x}^p - \boldsymbol{a}^p\|^2\right)^{1/2}$ = 1.2e-02

100 sensors and 27 anchors on $3 \times 3 \times 3$ grid in $[0,1]^3$

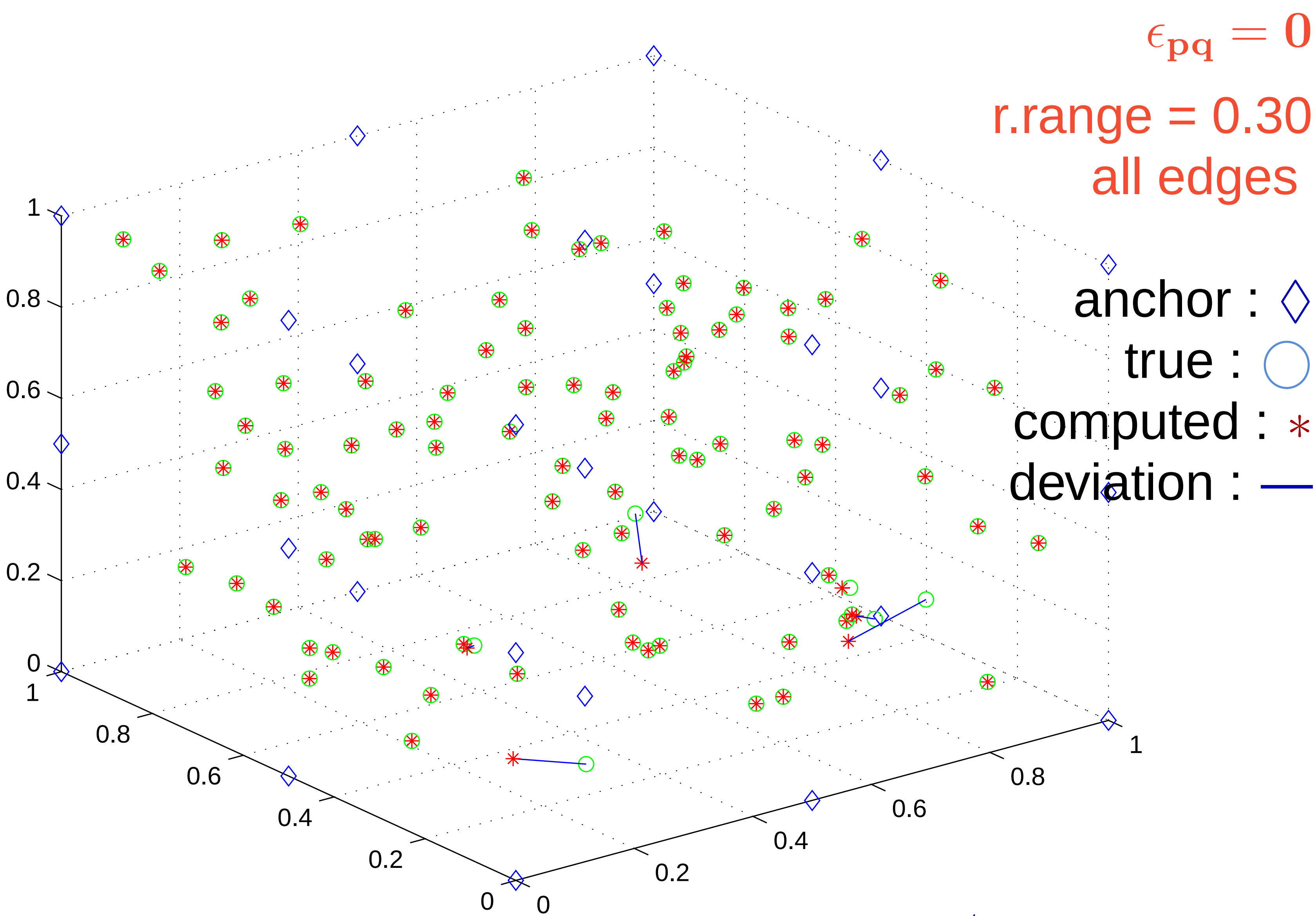
	$\epsilon_{pq} = 0$		$\epsilon_{pq} \sim 0.1 \times N(0,1)$	
radio range	rmsd	cpu	rmsd	cpu
0.25	2.8e-02	5.6	2.0e-02	10.3
0.30	3.4e-03	16.8	8.2e-03	19.8
0.30, all edges	3.4e-03	267.9		
0.35	3.7e-09	9.9	4.5e-03	14.4
0.40	2.2e-09	4.6	4.4e-03	11.8

cpu = SeDuMi cpu time in second ($\not\supset$ conversion time)

rmsd $= \frac{1}{m}\left(\sum_{p=1}^{m} \|\boldsymbol{x}^p - \boldsymbol{a}^p\|^2\right)^{1/2}$ (root mean square distance)

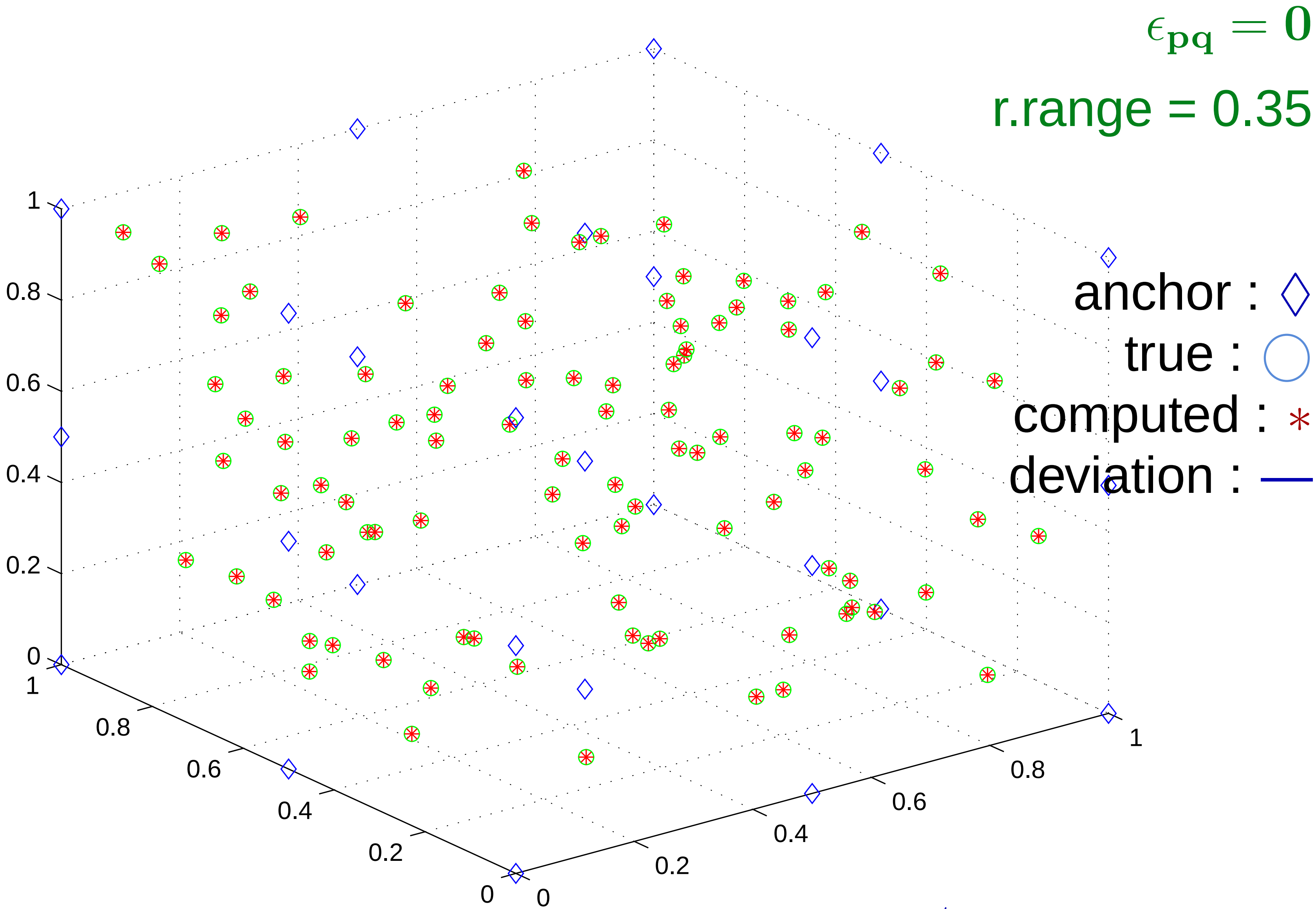
- radio range = 0.25, 0.30
  $\Rightarrow$ Not enough edges to determine all sensors' locations

100 sensors, 27 anchors, $3 \times 3 \times 3$ grid



267.9 sec., rmsd $= \frac{1}{m}\left(\sum_{p=1}^{m} \|\boldsymbol{x}^p - \boldsymbol{a}^p\|^2\right)^{1/2}$ = 3.4e-03

100 sensors, 27 anchors, $3 \times 3 \times 3$ grid

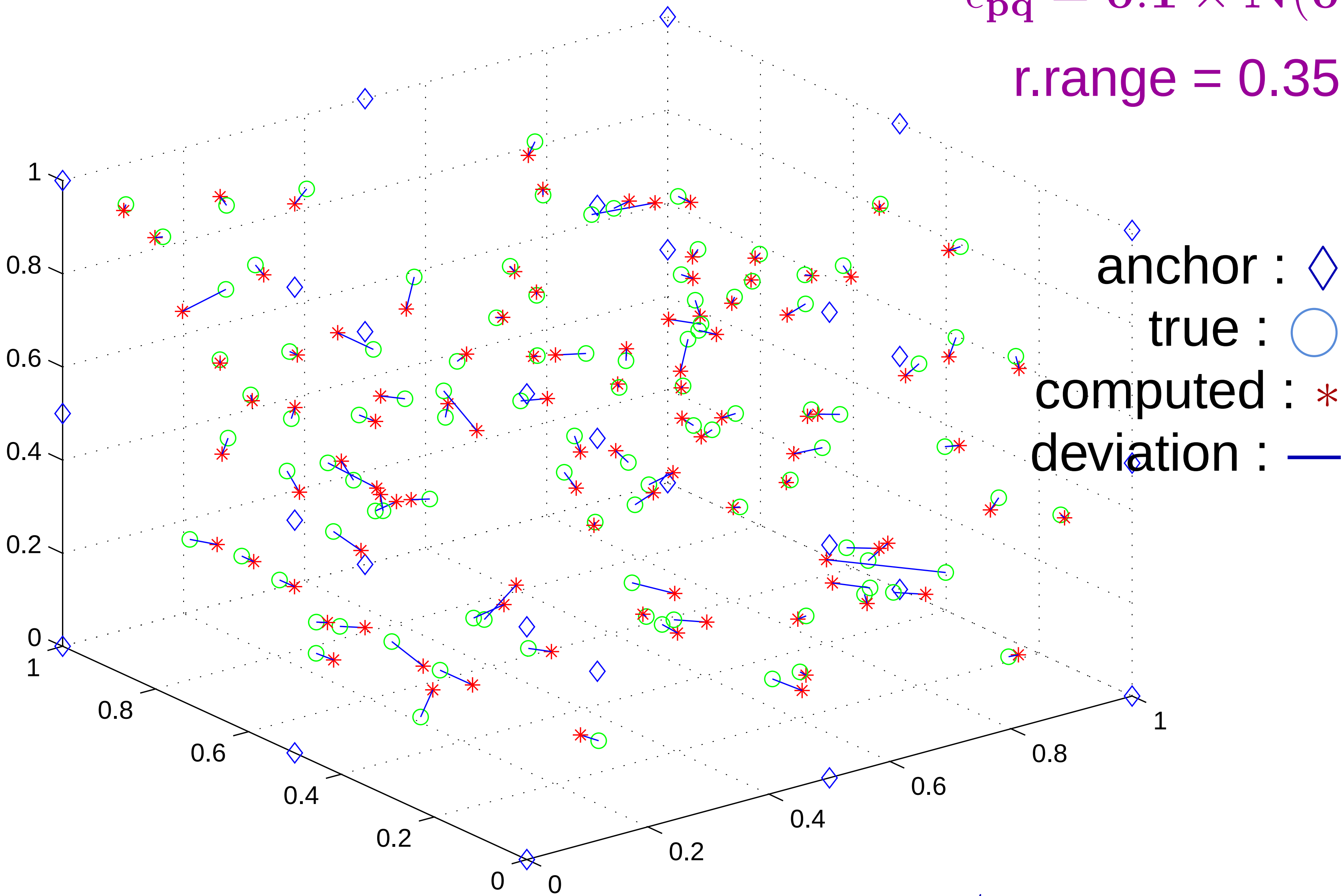


9.9 sec., rmsd $= \frac{1}{m}\left(\sum_{p=1}^{m} \|\boldsymbol{x}^p - \boldsymbol{a}^p\|^2\right)^{1/2}$ = 3.7e-09

100 sensors, 27 anchors, $3 \times 3 \times 3$ grid, $\epsilon_{pq} = 0.1 \times N(0,1)$

$\boldsymbol{\epsilon_{pq} = 0.1 \times N(0,1)}$

r.range = 0.35



14.4 sec., rmsd $= \frac{1}{m}\left(\sum_{p=1}^{m} \|\boldsymbol{x}^p - \boldsymbol{a}^p\|^2\right)^{1/2}$ = 4.5e-03

Contents

Concluding remarks

- Sparse SDP relaxation (Waki-Kim-Kojima-Muramatsu)
  = Lasserre's (dense) SDP relaxation + c-sparsity
  — poweful in practice and
  theoretical convergence (Lasserre)
- Thee remain many issues to be studied.
  - Exploiting sparsity further to solve larger scale POPs.
  - Large-scale SDPs.
  - Numerical difficulty in solving SDP relaxations of POPs.
  - Practically effective SDP relaxation for Polynomial SDPs.

Concluding remarks

- Sparse SDP relaxation (Waki-Kim-Kojima-Muramatsu)
  = Lasserre's (dense) SDP relaxation + c-sparsity
  — poweful in practice and
  theoretical convergence (Lasserre)
- Thee remain many issues to be studied.
  - Exploiting sparsity further to solve larger scale POPs.
  - Large-scale SDPs.
  - Numerical difficulty in solving SDP relaxations of POPs.
  - Practically effective SDP relaxation for Polynomial SDPs.

Thank you!