Canonical primal–dual algorithm for solving fourth-order polynomial minimization problems

Xiaojun Zhou [a,b], David Yang Gao [a,c,*], Chunhua Yang [b]

[a] School of Science, Information Technology and Engineering, University of Ballarat, Victoria 3353, Australia
[b] School of Information Science and Engineering, Central South University, Changsha 410083, China
[c] Research School of Engineering, The Australian National University, Canberra, ACT 0200, Australia

ARTICLE INFO

ABSTRACT

This paper focuses on implementation of a general canonical primal–dual algorithm for solving a class of fourth-order polynomial minimization problems. A critical issue in the canonical duality theory has been addressed, i.e., in the case that the canonical dual problem has no interior critical point in its feasible space $\mathcal{S}_a^+$, a quadratic perturbation method is introduced to recover the global solution through a primal–dual iterative approach, and a gradient-based method is further used to refine the solution. A series of test problems, including the benchmark polynomials and several instances of the sensor network localization problems, have been used to testify the effectiveness of the proposed algorithm.

Crown Copyright © 2013 Published by Elsevier Inc. All rights reserved.

1. Introduction

Polynomial optimization problems have been studied extensively with a wide range of applications in science and engineering, including chaotical dynamical systems, chemical database analysis, computational large deformation mechanics, network communication, and information science, etc. [6,9,10,12,13,17,18,21]. It is known that the minimization of a polynomial function on $\mathbb{R}^n$ is difficult even for fourth-order polynomials. Various methods have been presented to solve this kind of problems. By considering the first order conditions, algebraic techniques can be used for determining the real solutions to this system of polynomial equations; however, it is computationally expensive and the number of critical points can be infinite sometimes. Perturbation and relaxation techniques have attracted a lot of attention from diverse directions in recent decades. In [9], by perturbing the original polynomial based on even-order and combining it with semidefinite programming, a semidefinite approximation method was proposed for global unconstrained polynomial optimization. Semidefinite programming (SDP), second-order cone programming (SOCP) and sum of squares (SOS) relaxations are some of the most popular methods studied recently. The sequences of generalized Lagrangian dual and their SOS were introduced in [10]. In [18], based on the correlative sparsity pattern graph, sets of supports for sums of squares polynomials that lead to efficient SOS and SDP relaxations were obtained. A SOCP relaxation was studied in [17], and it showed that the SOCP relaxation was weaker than the SDP relaxation but had nice properties which made it useful as a problem preprocessor.

Canonical duality theory developed from nonconvex analysis and global optimization is an effective methodology [5], which has been used successfully for solving a large class of challenging problems in various disciplines, such as computational biology [20], large deformation mechanics [8,15], FIR filter design [19], and nonlinear dynamical systems [14]. In [6], the canonical duality theory for the sum of fourth-order polynomials minimization problems was established, and it showed that if the dual solution is in the feasible space $\mathcal{S}_a^+$, the corresponding primal solution can be obtained easily by the *canonical*

---

* Corresponding author at: School of Science, Information Technology and Engineering, University of Ballarat, Victoria 3353, Australia.
  E-mail address: david.gao@anu.edu.au (D.Y. Gao).

equilibrium equation. In this paper, we present a general canonical primal–dual algorithm for more challenging problems where, the canonical dual may have no critical point in $\mathcal{S}_a^+$. A quadratic perturbation approach is introduced to further optimize the problem through a primal–dual iterative method, and then a gradient-based algorithm is continued to refine the solution. The canonical primal–dual algorithm is tested on a set of benchmark polynomials and some instances of a sensor network localization problem. Experimental results have testified the effectiveness of the proposed algorithm.

2. Problem and canonical duality theory

For the completeness of this paper, we first present a brief review on the canonical dual methodology for solving the following fourth-order polynomials minimization problem (primal problem for short)

$$(\mathcal{P}) : \min\left\{P(\boldsymbol{x}) = W(\boldsymbol{x}) + \frac{1}{2}\boldsymbol{x}^T Q\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{f} : \boldsymbol{x} \in \mathbb{R}^n\right\}, \tag{1}$$

where,

$$W(\boldsymbol{x}) = \sum_{k=1}^m \frac{1}{2}\alpha_k\left(\frac{1}{2}\boldsymbol{x}^T A_k\boldsymbol{x} + \boldsymbol{b}_k^T\boldsymbol{x} + c_k\right)^2 \tag{2}$$

and $A_k = A_k^T, Q = Q^T \in \mathbb{R}^{n\times n}$ are indefinite symmetrical matrices, $\boldsymbol{b}_k, \boldsymbol{f} \in \mathbb{R}^n$ are given vectors, $\alpha_k, c_k \in \mathbb{R}$ are known constants. Without loss of much generality, the α_k is assumed to be positive.

By introducing a nonlinear operator

$$\xi = (\xi_1, \ldots, \xi_m)^T = \Lambda(\boldsymbol{x}) = \left\{\frac{1}{2}\boldsymbol{x}^T A_k\boldsymbol{x} + \boldsymbol{b}_k^T\boldsymbol{x} + c_k\right\}^m : \mathbb{R}^n \to \mathcal{E}_a \subset \mathbb{R}^m, \tag{3}$$

the nonconvex function $W(\boldsymbol{x})$ can be recast by

$$W(\boldsymbol{x}) = V(\Lambda(\boldsymbol{x})), \tag{4}$$

where, $V(\xi)$ is a quadratic function defined by

$$V(\xi) = \sum_{k=1}^m \frac{1}{2}\alpha_k\xi_k^2 = \frac{1}{2}\boldsymbol{\alpha}^T(\xi \circ \xi), \tag{5}$$

in which, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)^T$, the notation $\boldsymbol{s} \circ \boldsymbol{t} = (s_1 t_1, \ldots, s_m t_m)^T$ denotes the Hadamard product for any two vectors $\boldsymbol{s}, \boldsymbol{t} \in \mathbb{R}^m$.

Remark 1. According to the canonical duality theory, the quadratic function $V(\xi)$ is said to be a canonical function if its domain $\mathcal{E}_a$ is a convex set. This requires that the nonlinear mapping $\Lambda : \mathbb{R}^n \to \mathcal{E}_a$ should be a so-called *geometrically admissible measure*. Particularly, $\Lambda(\boldsymbol{x})$ is an *objective measure* if $A_k \succeq 0$, $\boldsymbol{b}_k = 0$, $\forall k = 1, \ldots, m$ (see Definition 6.1.2 in [4] for the objectivity in continuum physics).

In this paper, we assume that $\Lambda(\boldsymbol{x})$ is simply a geometrical measure such that $\mathcal{E}_a$ is a convex set in $\mathbb{R}^m$. Then, the primal problem can be rewritten as the canonical form:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n}\{P(\boldsymbol{x}) = V(\Lambda(\boldsymbol{x})) - U(\boldsymbol{x})\}, \tag{6}$$

where, $U(\boldsymbol{x}) = -\frac{1}{2}\boldsymbol{x}^T Q\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{f}$.

The dual variable ς to ξ is defined by the duality mapping

$$\varsigma = (\varsigma_1, \ldots, \varsigma_m) = \nabla V(\xi) = \boldsymbol{\alpha} \circ \xi : \mathcal{E}_a \to \mathcal{E}_a^* \subset \mathbb{R}^m. \tag{7}$$

For the given canonical function $V(\xi)$, its Legendre conjugate $V^*(\varsigma)$ can be defined by:

$$V^*(\varsigma) = \underset{\xi}{\mathrm{sta}}\{\xi^T\varsigma - V(\xi)\} = \sum_{k=1}^m \frac{1}{2}\alpha_k^{-1}\varsigma_k^2, \tag{8}$$

where, $\mathrm{sta}\{\cdot\}$ stands for finding stationary point of the statement in $\{\cdot\}$. The (ξ, ς) forms a canonical duality pair and the following canonical duality relations hold on $\mathcal{E}_a \times \mathcal{E}_a^*$:

$$\varsigma = \nabla V(\xi) \iff \xi = \nabla V^*(\varsigma) \iff V(\xi) + V^*(\varsigma) = \xi^T\varsigma. \tag{9}$$

Obviously, we have the following reverse Legendre conjugate

$$V(\Lambda(\boldsymbol{x})) = \max\{\Lambda(\boldsymbol{x})^T\varsigma - V^*(\varsigma)|\varsigma \in \mathcal{E}_a^*\}. \tag{10}$$

By substituting (10) into (6), problem $(\mathcal{P})$ can be equivalent to

$$\min_{\boldsymbol{x}}\max_{\varsigma}\{\Xi(\boldsymbol{x}, \varsigma)|\boldsymbol{x} \in \mathbb{R}^n, \varsigma \in \mathcal{E}_a^*\}, \tag{11}$$

where, $\Xi(\boldsymbol{x},\varsigma)$ is the *generalized (or total) complementary function* defined by

$$\Xi(\boldsymbol{x},\varsigma) = \Lambda^T(\boldsymbol{x})\varsigma - V^*(\varsigma) - U(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T G(\varsigma)\boldsymbol{x} - \boldsymbol{x}^T F(\varsigma) + \sum_{k=1}^{m}\left(c_k\varsigma_k - \frac{1}{2}\alpha_k^{-1}\varsigma_k^2\right), \tag{12}$$

here,

$$G(\varsigma) = Q + \sum_{k=1}^{m}\varsigma_k A_k, \quad F(\varsigma) = \boldsymbol{f} - \sum_{k=1}^{m}\varsigma_k \boldsymbol{b}_k. \tag{13}$$

For a given $\varsigma \in \mathcal{E}_a^*$, the criticality condition $\nabla_{\boldsymbol{x}}\Xi(\boldsymbol{x},\varsigma)$ leads to the *canonical equilibrium equation*:

$$G(\varsigma)\boldsymbol{x} = F(\varsigma). \tag{14}$$

Solving this linear equation for $\boldsymbol{x}$ and substituting the result back to the total complementary function $\Xi(\boldsymbol{x},\varsigma)$, we obtain the canonical dual function

$$P^d(\varsigma) = \sum_{k=1}^{m}\left(c_k\varsigma_k - \frac{1}{2}\alpha_k^{-1}\varsigma_k^2\right) - \frac{1}{2}F^T(\varsigma)G^{-1}(\varsigma)F(\varsigma). \tag{15}$$

By introducing the following canonical dual feasible space

$$\mathcal{S}_a^+ = \{\varsigma \in \mathcal{E}_a^* | G(\varsigma) \succ 0\}, \tag{16}$$

the canonical dual problem can be formulated by

$$(\mathcal{P}^d): \ \max\left\{P^d(\varsigma): \ \varsigma \in \mathcal{S}_a^+\right\}. \tag{17}$$

Theorem 1. *The problem $(\mathcal{P}^d)$ is canonically dual to the primal problem $(\mathcal{P})$ in the sense that if $\bar\varsigma$ is a critical point of $(\mathcal{P}^d)$, then the vector*

$$\bar{\boldsymbol{x}} = G^{-1}(\bar\varsigma)F(\bar\varsigma) \tag{18}$$

is a critical point of $(\mathcal{P})$ and

$$P(\bar{\boldsymbol{x}}) = P^d(\bar\varsigma). \tag{19}$$

Furthermore, If $\bar\varsigma \in \mathcal{S}_a^+$, then $\bar{\boldsymbol{x}}$ is a global minimizer of $(\mathcal{P})$ if and only if $\bar\varsigma$ is a global maximizer of $(\mathcal{P}^d)$, i.e.,

$$P(\bar{\boldsymbol{x}}) = \min_{\boldsymbol{x}\in\mathbb{R}^n}P(\boldsymbol{x}) \Longleftrightarrow \max_{\varsigma\in\mathcal{S}_a^+}P^d(\varsigma) = P^d(\bar\varsigma). \tag{20}$$

Proof. The first part of this theorem is the so-called complementary-dual principle proposed by Gao in [3]. Here we need to prove the convexity of $\mathcal{S}_a^+$. For any given $\varsigma_1, \varsigma_2 \in \mathcal{S}_a^+$, we should have

$$\theta G(\varsigma_1) \succ 0, \quad (1-\theta)G(\varsigma_2) \succ 0 \quad \forall \theta \in [0,1].$$

Therefore,

$$\theta G(\varsigma_1) + (1-\theta)G(\varsigma_2) = G(\theta\varsigma_1 + (1-\theta)\varsigma_2) \succ 0 \quad \forall \theta \in [0,1].$$

This shows that $\mathcal{S}_a^+$ is convex.

By the fact that the total complementary function $\Xi(\mathbf{x},\varsigma)$ is a saddle function on $\mathbb{R}^n \times \mathcal{S}_a^+$, the classical saddle min–max duality theory (cf. [2, page 57] or [4, page 39]) leads to

$$\min P(\mathbf{x}) = \min_{\mathbf{x}\in\mathbb{R}^n}\max_{\varsigma\in\mathcal{S}_a^+}\Xi(\mathbf{x},\varsigma) = \max_{\varsigma\in\mathcal{S}_a^+}\min_{\mathbf{x}\in\mathbb{R}^n}\Xi(\mathbf{x},\varsigma) = \max_{\varsigma\in\mathcal{S}_a^+}P^d(\varsigma).$$

Therefore, by the complementary-dual principle, the critical point $\bar{\mathbf{x}} \in \mathbb{R}^n$ of Ξ is a global min of $P(\mathbf{x})$ if and only if the associated critical point $\bar\varsigma$ is a global max of $P^d(\varsigma)$ on $\mathcal{S}_a^+$. Since $\Xi(\mathbf{x},\varsigma)$ is strictly convex in $\mathbf{x}$ and concave in ς on $\mathbb{R}^n \times \mathcal{S}_a^+$, its saddle point is unique. $\square$

The statement (20) is the so-called canonical min–max duality, which is the first part of the *triality theory* originally discovered in Gao and Strang's 1989 paper [7] theorem shows that the canonical dual problem is a concave maximization over a convex set and the condition $G(\varsigma) \succ 0$ in $\mathcal{S}_a^+$ provides a global optimality criterion for the nonconvex primal problem. By the fact that the canonical dual function $P^d(\varsigma)$ is concave on the convex domain $\mathcal{S}_a^+$, this canonical dual can be solved easily by well-developed nonlinear optimization techniques, say, the SDP method. To see this, we first relax $(\mathcal{P}^d)$ to

$$\min \{g_1 + g_2 - \varsigma^T \boldsymbol{c}\},$$

$$\text{s.t. } g_1 \geqslant \frac{1}{2} F^T(\varsigma) G^{-1}(\varsigma) F(\varsigma),$$

$$g_2 \geqslant \frac{1}{2} \varsigma^T [\text{Diag}\{\alpha_1, \ldots, \alpha_m\}]^{-1} \varsigma,$$

$$G(\varsigma) \succeq 0. \tag{21}$$

Then, by using the Schur complement Lemma [1], we have

$$\min \{g_1 + g_2 - \varsigma^T \boldsymbol{c}\},$$

$$\text{s.t. } \begin{pmatrix} G(\varsigma) & F(\varsigma) \\ F^T(\varsigma) & 2g_1 \end{pmatrix} \succeq 0,$$

$$\begin{pmatrix} \text{Diag}\{\alpha_1, \ldots, \alpha_m\} & \varsigma \\ \varsigma^T & 2g_2 \end{pmatrix} \succeq 0, \tag{22}$$

which is a typical semi-definite programming problem.

If the canonical dual solution is an interior point of $\mathcal{S}_a^+$, then it must be a critical point of $P^d(\varsigma)$ and the associated $\bar{\boldsymbol{x}} = G^{-1}(\bar{\varsigma})F(\bar{\varsigma})$ is the unique global minimizer of the primal problem. Otherwise, this $\bar{\boldsymbol{x}}$ is only a lower bound solution to the primal problem.

3. Canonical primal–dual algorithm

In this section, we shall address a critical question in the canonical duality theory: how to solve the nonconvex problem $(\mathcal{P})$ if its canonical dual problem $(\mathcal{P}^d)$ has no interior critical point in $\mathcal{S}_a^+$? Clearly, if $P^d(\varsigma)$ has no critical point $\bar{\varsigma} \in \text{int } \mathcal{S}_a^+$, its critical point could be either on the boundary of $\mathcal{S}_a^+$ or located in the out side of $\mathcal{S}_a^+$. In order to solve this problem, we use the quadratic perturbation method by introducing a positive parameter $\delta_k > 0$ and let

$$\Xi_{\delta_k}(\boldsymbol{x}, \varsigma) = \Xi(\boldsymbol{x}, \varsigma) + \frac{\delta_k}{2} \|\boldsymbol{x} - \boldsymbol{x}_k\|^2 = \frac{1}{2} \boldsymbol{x}^T G_{\delta_k}(\varsigma) \boldsymbol{x} - \boldsymbol{x}^T F_{\delta_k}(\varsigma) + \sum_{k=1}^{m} \left(c_k \varsigma_k - \frac{1}{2} \alpha_k^{-1} \varsigma_k^2 \right) + \frac{\delta_k}{2} \boldsymbol{x}_k^T \boldsymbol{x}_k,$$

where,

$$G_{\delta_k}(\varsigma) = G(\varsigma) + \delta_k I, \quad F_{\delta_k}(\varsigma) = F(\varsigma) + \delta_k \boldsymbol{x}_k.$$

Thus, the original dual feasible space $\mathcal{S}_a^+$ can be relaxed to

$$\mathcal{S}_{\delta_k}^+ = \{\varsigma \in \mathcal{E}_a^* | G_{\delta_k}(\varsigma) \succ 0\}.$$

Clearly, we have $\mathcal{S}_a^+ \subset \mathcal{S}_{\delta_k}^+ \ \forall \delta_k > 0$. Therefore, for each given δ_k and $\boldsymbol{x}_k$, the perturbed canonical dual problem can be proposed as

$$(\mathcal{P}_{\delta_k}^d) : \max_{\varsigma \in \mathcal{S}_{\delta_k}^+} \left\{ P_{\delta_k}^d(\varsigma) = \sum_{k=1}^{m} \left(c_k \varsigma_k - \frac{1}{2} \alpha_k^{-1} \varsigma_k^2 \right) - \frac{1}{2} F_{\delta_k}^T(\varsigma) G_{\delta_k}^{-1}(\varsigma) F_{\delta_k}(\varsigma) \right\}. \tag{23}$$

Based on this perturbed problem, the following canonical primal–dual algorithm can be proposed for solving the nonconvex problem $(\mathcal{P})$.

Algorithm 1 (*Canonical primal–dual algorithm*). Given initial data $\delta_0 > 0$, $\boldsymbol{x}_0 \in \mathbb{R}^n$, and error allowance $\epsilon > 0$, let $k = 0$.

1. Solve the perturbed canonical dual problem $(\mathcal{P}_{\delta_k}^d)$ to obtain $\varsigma_k \in S_{\delta_k}^+$.
2. Compute $\widetilde{\boldsymbol{x}}_{k+1} = [G_{\delta_k}(\varsigma_k)]^{-1} F_{\delta_k}(\varsigma_k)$ and let
   $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \beta_k (\widetilde{\boldsymbol{x}}_{k+1} - \boldsymbol{x}_k), \ \beta_k \in [0, 1]$.
3. If $|P(\boldsymbol{x}_{k+1} - P(\boldsymbol{x}_k)| \leqslant \epsilon$, then stop, $\boldsymbol{x}_{k+1}$ is the optimal solution. Otherwise, let $k = k + 1$, go back to step 1.

In this algorithm, $\{\beta_k\} \in [0, 1]$ are given parameters, which change the search directions. Clearly, if $\beta_k = 1$, we have $\boldsymbol{x}_{k+1} = \widetilde{\boldsymbol{x}}_{k+1}$.

Remark 2. It is easy to understand that if the perturbation parameter $\delta_k \to 0$, then $\Xi_{\delta_k}(\boldsymbol{x}, \varsigma) \to \Xi(\boldsymbol{x}, \varsigma)$. This shows that the goal of the quadratic perturbation method is to find an alternative solution in the neighborhood of $\boldsymbol{x}_k$.

Based on this algorithm, the whole pseudocode of the canonical primal–dual algorithm can be described as follows:

1: $\varsigma_{old} \leftarrow$ canonical_dual $(\alpha, A, b, c, Q, f, m, n)$
2: **if** $G(\varsigma_{old}) \succ 0$ **then**
3:    $\boldsymbol{x}_{old} \leftarrow G^{-1}(\varsigma_{old})F(\varsigma_{old})$
4:    $P(x_{old}) \leftarrow$ canonical_primal$(\boldsymbol{x}_{old}, \alpha, A, b, c, Q, f, m, n)$
5: **else**
6:   **repeat**
7:      $\varsigma_{new} \leftarrow$ canonical_dual$(\delta_k, \boldsymbol{x}_{old}, \alpha, A, b, c, Q, f, m, n)$
8:      $\boldsymbol{x}_{new} \leftarrow G_{\delta_k}^{-1}(\varsigma_{new})F_{\delta_k}(\varsigma_{new})$
9:      $P(x_{new}) \leftarrow$ canonical_primal$(\boldsymbol{x}_{new}, \alpha, A, b, c, Q, f, m, n)$
10:     **if** $\|P(x_{new}) - P(x_{old})\| < \epsilon$ or $\|\boldsymbol{x}_{new} - \boldsymbol{x}_{old}\| < \epsilon$ **then**
11:     **break**
12:     **else**
13:      $\delta_k \leftarrow \delta_k/2$
14:      $\boldsymbol{x}_{old} \leftarrow \boldsymbol{x}_{new}$
15:      $P(x_{old}) \leftarrow P(x_{new})$
16:     **end if**
17:   **until** the maximum number of iterations is met
18: **end if**
19: $[\boldsymbol{x}_{out}, P(x_{out})] \leftarrow$ gradient_based$(@(x)$canonical_primal$(x, \alpha, A, b, c, Q, f, m), \boldsymbol{x}_{new})$

---

where, $\alpha(1) = \alpha_1, \ldots, \alpha(m) = \alpha_m$; $A\{1\} = A_1, \ldots, A\{m\} = A_m$; $b\{1\} = \boldsymbol{b}_1, \ldots, b\{m\} = \boldsymbol{b}_m$; $c(1) = c_1, \ldots, c(m) = c_m$; the function *canonical_primal* is to calculate the objectives of the primal problem $(\mathcal{P})$; the function *canonical_dual* is for the implementation of solving the canonical dual problem $(\mathcal{P}^d)$ or $(\mathcal{P}^d_{\delta_k})$, and *gradient_based* can be regarded as the refinement of the proximal point method.

4. Application for benchmark problems

The proposed canonical primal–dual algorithm is implemented in MATLAB R2010b on Intel (R) Core (TM) i3-2310 M CPU @2.10 GHz under Window 7 environment. For the SDP subproblem, a software package named SeDuMi [16] is used in this study, and the built-in function *fminunc* in MATLAB is adopted to represent the gradient-based method.

4.1. Benchmark functions

We first list some of well-known benchmark functions:
Colville function:

$$f_1 = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1).$$

Zettle function:

$$f_2 = (x_1^2 + x_2^2 - 2x_1)^2 + 0.25x_1.$$

Extended Styblinski–Tang function:

$$f_3 = \frac{1}{2}\sum_{i=1}^{n}(x_i^4 - 16x_i^2 + 5x_i).$$

Rosenbrock function:

$$f_4 = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2].$$

Dixon & Price function:

$$f_5 = (x_1 - 1)^2 + \sum_{i=2}^{n}i(2x_i^2 - x_{i-1})^2.$$

4.2. Experimental results

Test results for the Colville and Zettle problems are given in Table 1. We can find that the iterations are both 1, which indicates that these two problems can be solved directly without the proximal point method. Table 2 gives the experimental

Table 1
Experimental results for the Colville and Zettle problems.

Functions	x^*	$P(x^*)$	Iterations	Time (s)
f_1	$(1,1,1,1)$	$1.7699e{-}10$	1	0.640260
f_2	$(-0.0299,0)$	-0.0038	1	0.649753

Table 2
Experimental results for the Extended Styblinski–Tang problem.

n	x^*	$P(x^*)$	Iterations	Time (s)
2	$(-2.9035,-2.9035)$	-78.3323	1	0.606217
5	$(-2.9035,\ldots,-2.9035)$	-195.8308	1	0.666961
10	$(-2.9035,\ldots,-2.9035)$	-391.6617	1	0.669273
20	$(-2.9035,\ldots,-2.9035)$	-783.3233	1	0.777645
50	$(-2.9035,\ldots,-2.9035)$	$-1.9583e3$	1	1.441296
100	$(-2.9035,\ldots,-2.9035)$	$-3.9166e3$	1	4.709524

Fig. 1. Computational results for the Extended Styblinski–Tang function ($n = 20$).

results for the Extended Styblinski–Tang problem, and Fig. 1 gives the output results for the problem when $n = 20$. We can find that if the size of problem is less than 10, the difference of time consuming is trivial. Experimental results for the Rosenbrock and Dixon & Price problems are given in Tables 3 and 4, respectively. It is easy to find that when the size of problem is small, few proximal point iterations are needed, but it requires more iterations when the size is larger.

Table 3
Experimental results for the Rosenbrock problem.

n	$\boldsymbol{x}^*$	$P(\boldsymbol{x}^*)$	Iterations	Time (s)
2	$(1,1)$	2.0051e−11	2	0.705291
5	$(1,\ldots,1)$	1.1091e−11	2	0.738990
10	$(1,\ldots,1)$	5.8661e−11	2	0.872011
20	$(1,\ldots,1)$	6.2422e−10	2	1.222506
50	$(1,\ldots,1)$	8.2778e−11	2	3.315423
100	$(1,\ldots,1)$	9.3099e−8	3	9.855752

Table 4
Experimental results for the Dixon & Price problem.

n	$\boldsymbol{x}^*$	$P(\boldsymbol{x}^*)$	Iterations	Time (s)
2	$(1,0.7071)$	4.7740e−15	2	0.658417
5	$(1,0.7071,0.5946,0.5453,0.5221)$	3.9968e−15	2	0.729285
10	$(1,0.7071,\ldots,0.5014,0.5007)$	5.1454e−12	2	0.773957
20	$(1,0.7071,\ldots,0.5000,0.5000)$	8.3509e−10	3	1.063605
50	$(1,0.7071,\ldots,0.5000,0.5000)$	1.0772e−10	3	6.032329
100	$(1,0.7071,\ldots,0.5000,0.5000)$	6.0383e−9	3	15.031212
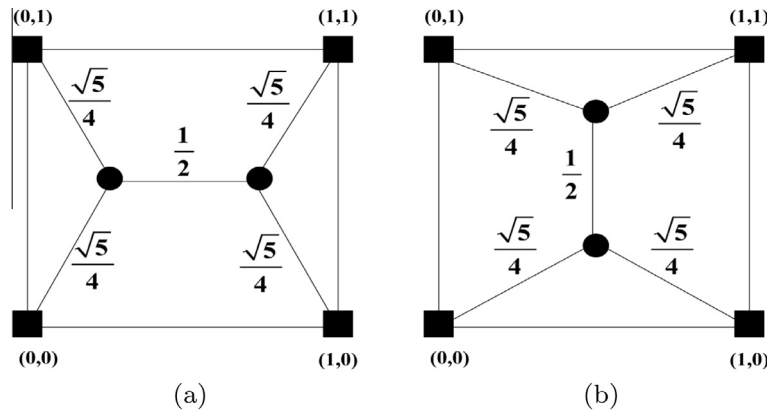
Fig. 2. Network topology for the motivating examples.

5. Application for the sensor network localization

The sensor network localization problem can be described as follows. Given m anchor points $\boldsymbol{a}_1,\cdots,\boldsymbol{a}_m \in \mathbb{R}^d$ (d is usually 2 or 3), the distance d_{ij} (Euclidean distance) between the ith and jth anchor points if $(i,j) \in N_x$, and the distance e_{ik} between the ith sensor and kth anchor points if $(i,k) \in N_a$, where, $N_x = \{(i,j) : \|\boldsymbol{x}_i - \boldsymbol{x}_j\| = d_{ij} \leqslant r_d\}$ and $N_a = \{(i,k) : \|\boldsymbol{x}_i - \boldsymbol{a}_k\| = e_{ik} \leqslant r_d\}$, here, r_d is the radio range, the sensor network localization problem is to find n distinct sensor points $\boldsymbol{x}_i,\ i = 1,\ldots,n$, such that

$$\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 = d_{ij}^2, \quad \forall (i,j) \in N_x,$$
$$\|\boldsymbol{x}_i - \boldsymbol{a}_k\|^2 = e_{ik}^2, \quad \forall (i,k) \in N_a. \tag{24}$$

Usually, the distance d_{ij}, e_{ij} may contain noise, leading the Eq. (24) infeasible; therefore, using the least squares method, we can formulate it to the following nonconvex optimization problem

$$\min_{\boldsymbol{x}_i,\ldots,\boldsymbol{x}_n} \sum_{(i,j) \in N_x} (\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 - d_{ij}^2)^2 + \sum_{(i,k) \in N_a} (\|\boldsymbol{x}_i - \boldsymbol{a}_k\|^2 - e_{ik}^2)^2. \tag{25}$$

Suppose that the number of existing edges between every pair of two sensor points, and between every pair of anchor and sensor points are n_x and n_a, respectively. Denote $\boldsymbol{x} = [x_{11},x_{12},\ldots,x_{i1},x_{i2},\ldots,x_{n1},x_{n2}]^T \in \mathbb{R}^{2n}$ ($d = 2$), then

$$(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 - d_{ij}^2) = (x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 - d_{ij}^2 = \frac{1}{2}\boldsymbol{x}^T A_s \boldsymbol{x} + \boldsymbol{b}_s^T \boldsymbol{x} + c_s$$

Table 5
Known data for the motivating examples.

Instances	Anchor points	Distances
(a)	$\boldsymbol{a}_1 = (0,0)$, $\boldsymbol{a}_2 = (0,1)$, $\boldsymbol{a}_3 = (1,0)$, $\boldsymbol{a}_4 = (1,1)$	$d_{12} = 0.5$, $e_{11} = \frac{\sqrt{5}}{4}$, $e_{12} = \frac{\sqrt{5}}{4}$, $e_{23} = \frac{\sqrt{5}}{4}$, $e_{24} = \frac{\sqrt{5}}{4}$
(b)	$\boldsymbol{a}_1 = (0,0)$, $\boldsymbol{a}_2 = (0,1)$, $\boldsymbol{a}_3 = (1,0)$, $\boldsymbol{a}_4 = (1,1)$	$d_{12} = 0.5$, $e_{11} = \frac{\sqrt{5}}{4}$, $e_{13} = \frac{\sqrt{5}}{4}$, $e_{22} = \frac{\sqrt{5}}{4}$, $e_{24} = \frac{\sqrt{5}}{4}$

Table 6
Experimental results for 10 sensors and 4 anchors.

True solutions	Solutions by the Proposed algorithm
$\boldsymbol{x}_1^* = (0.8179, 0.9675)$	$\bar{\boldsymbol{x}}_1 = (0.8184, 0.9668)$
$\boldsymbol{x}_2^* = (0.7907, 0.0915)$	$\bar{\boldsymbol{x}}_2 = (0.7903, 0.0913)$
$\boldsymbol{x}_3^* = (0.3421, 0.4704)$	$\bar{\boldsymbol{x}}_3 = (0.3416, 0.4705)$
$\boldsymbol{x}_4^* = (0.9365, 0.1694)$	$\bar{\boldsymbol{x}}_4 = (0.9363, 0.1693)$
$\boldsymbol{x}_5^* = (0.5058, 0.2568)$	$\bar{\boldsymbol{x}}_5 = (0.5060, 0.2576)$
$\boldsymbol{x}_6^* = (0.3008, 0.7197)$	$\bar{\boldsymbol{x}}_6 = (0.3007, 0.7201)$
$\boldsymbol{x}_7^* = (0.8456, 0.6711)$	$\bar{\boldsymbol{x}}_7 = (0.8456, 0.6709)$
$\boldsymbol{x}_8^* = (0.4286, 0.7408)$	$\bar{\boldsymbol{x}}_8 = (0.4284, 0.7407)$
$\boldsymbol{x}_9^* = (0.0698, 0.6631)$	$\bar{\boldsymbol{x}}_9 = (0.0691, 0.6624)$
$\boldsymbol{x}_{10}^* = (0.1757, 0.7560)$	$\bar{\boldsymbol{x}}_{10} = (0.1751, 0.7563)$

Table 7
Experimental results for 20 sensors and 4 anchors.

True solutions	Solutions by the Proposed algorithm
$\boldsymbol{x}_1^* = (0.8179, 0.9675)$	$\bar{\boldsymbol{x}}_1 = (0.8174, 0.9679)$
$\boldsymbol{x}_2^* = (0.7907, 0.0915)$	$\bar{\boldsymbol{x}}_2 = (0.7908, 0.0918)$
$\boldsymbol{x}_3^* = (0.3421, 0.4704)$	$\bar{\boldsymbol{x}}_3 = (0.3420, 0.4704)$
$\boldsymbol{x}_4^* = (0.9365, 0.1694)$	$\bar{\boldsymbol{x}}_4 = (0.9369, 0.1699)$
$\boldsymbol{x}_5^* = (0.5058, 0.2568)$	$\bar{\boldsymbol{x}}_5 = (0.5058, 0.2565)$
$\boldsymbol{x}_6^* = (0.3008, 0.7197)$	$\bar{\boldsymbol{x}}_6 = (0.3003, 0.7197)$
$\boldsymbol{x}_7^* = (0.8456, 0.6711)$	$\bar{\boldsymbol{x}}_7 = (0.8456, 0.6714)$
$\boldsymbol{x}_8^* = (0.4286, 0.7408)$	$\bar{\boldsymbol{x}}_8 = (0.4283, 0.7406)$
$\boldsymbol{x}_9^* = (0.0698, 0.6631)$	$\bar{\boldsymbol{x}}_9 = (0.0697, 0.6632)$
$\boldsymbol{x}_{10}^* = (0.1757, 0.7560)$	$\bar{\boldsymbol{x}}_{10} = (0.1759, 0.7560)$
$\boldsymbol{x}_{11}^* = (0.6441, 0.9545)$	$\bar{\boldsymbol{x}}_{11} = (0.6437, 0.9544)$
$\boldsymbol{x}_{12}^* = (0.5325, 0.2370)$	$\bar{\boldsymbol{x}}_{12} = (0.5325, 0.2374)$
$\boldsymbol{x}_{13}^* = (0.6727, 0.7358)$	$\bar{\boldsymbol{x}}_{13} = (0.6723, 0.7360)$
$\boldsymbol{x}_{14}^* = (0.1614, 0.2264)$	$\bar{\boldsymbol{x}}_{14} = (0.1612, 0.2268)$
$\boldsymbol{x}_{15}^* = (0.7012, 0.5950)$	$\bar{\boldsymbol{x}}_{15} = (0.7008, 0.5954)$
$\boldsymbol{x}_{16}^* = (0.8939, 0.4577)$	$\bar{\boldsymbol{x}}_{16} = (0.8939, 0.4582)$
$\boldsymbol{x}_{17}^* = (0.7856, 0.7407)$	$\bar{\boldsymbol{x}}_{17} = (0.7852, 0.7408)$
$\boldsymbol{x}_{18}^* = (0.7764, 0.4873)$	$\bar{\boldsymbol{x}}_{18} = (0.7759, 0.4876)$
$\boldsymbol{x}_{19}^* = (0.3702, 0.2146)$	$\bar{\boldsymbol{x}}_{19} = (0.3698, 0.2145)$
$\boldsymbol{x}_{20}^* = (0.1967, 0.2493)$	$\bar{\boldsymbol{x}}_{20} = (0.1966, 0.2495)$

where, $A_s \in \mathbb{R}^{2n \times 2n}$ is a sparse symmetric matrix with
$A_s(2i-1, 2i-1) = 2$, $A_s(2j-1, 2j-1) = 2$, $A_s(2i, 2i) = 2$, $A_s(2j, 2j) = 2$, $A_s(2i-1, 2j-1) = -2$,
$A_s(2j-1, 2i-1) = -2$, $A_s(2i, 2j) = -2$, $A_s(2j, 2i) = -2$, $\boldsymbol{b}_s = \boldsymbol{0}$, $c_s = -d_{ij}^2$.

$$\left(\|\boldsymbol{x}_i - \boldsymbol{a}_k\|^2 - e_{ik}^2 \right) = (x_{i1} - a_{k1})^2 + (x_{i2} - a_{k2})^2 - e_{ik}^2 = \frac{1}{2}\boldsymbol{x}^T A_t \boldsymbol{x} + \boldsymbol{b}_t^T \boldsymbol{x} + c_t$$

where, $A_t \in \mathbb{R}^{2n \times 2n}$ is a sparse symmetric matrix with

$$A_t(2i-1, 2i-1) = 2, \quad A_t(2i, 2i) = 2,$$

$\boldsymbol{b}_t \in \mathbb{R}^{2n}$ is a sparse vector with

$$b_t(2i-1, 1) = -2a_{k1}, \quad b_t(2i, 1) = -2a_{k2}$$
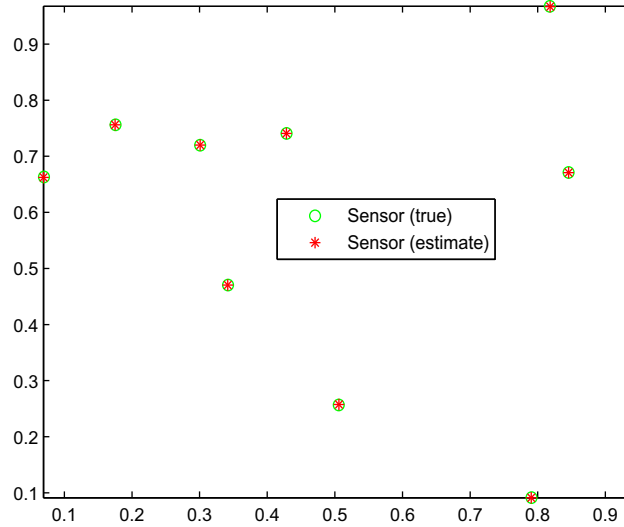
and $c_t = a_{k1}^2 + a_{k2}^2 - e_{ik}^2$.

Fig. 3. Computed locations of 10 sensors and 4 anchors.
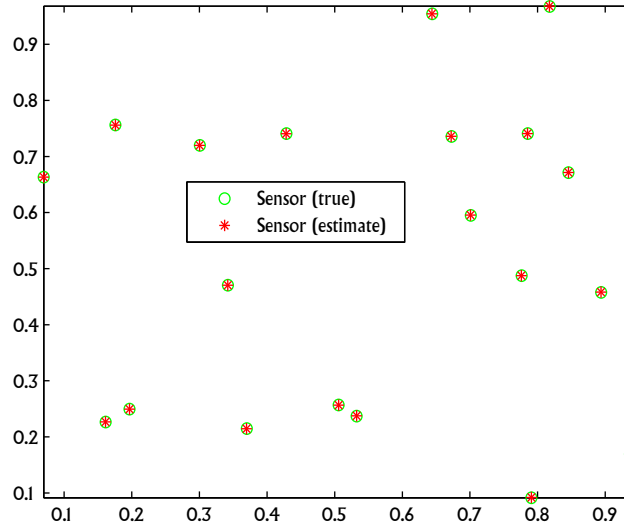

Fig. 4. Computed locations of 20 sensors and 4 anchors.

Then the optimization problem can be unified as the fourth-order type:

$$\min_{\mathbf{x}} \sum_{s=1}^{n_x} \left(\frac{1}{2}\mathbf{x}^T A_s \mathbf{x} + \mathbf{b}_s^T \mathbf{x} + c_s\right)^2 + \sum_{t=1}^{n_a} \left(\frac{1}{2}\mathbf{x}^T A_t \mathbf{x} + \mathbf{b}_t^T \mathbf{x} + c_t\right)^2.$$

5.1. Motivating examples

Considering the following two motivating examples with the network topology given in Fig. 2(a) and (b), respectively. The data for these problems are given in Table 5.

By using the proposed canonical primal–dual algorithm, for both examples, in the 1st iteration, we can get $\bar{\varsigma} = 1.0\mathrm{e}-5(0.1538,\ldots,0.1538)^T$. Through Cholesky factorization, we can find that $G_i(\bar{\varsigma}) \succ 0$ $(i = 1, 2)$ is positive definite, and the corresponding $\bar{\mathbf{x}}_1 = G_1^{-1}(\bar{\varsigma})F_1(\bar{\varsigma}) = (0.25, 0.5, 0.75, 0.5)^T, \bar{\mathbf{x}}_2 = G_2^{-1}(\bar{\varsigma})F_2(\bar{\varsigma}) = (0.5, 0.25, 0.5, 0.75)^T$.

Remark 3. The $\bar{\varsigma}$ is small but valid due to the perturbed complementary slackness in primal–dual interior point method used in SeDuMi.

5.2. Other instances

Two other instances are randomly created by SFSDP [11], a MATLAB package for solving sensor network localization problems. The first one is a 10 sensors and 4 anchors problem with the radio range at 1, and the second problem is a 20 sensors and 4 anchors problem with the radio range at 0.5. For both problems, the positions of 4 anchors are the same, and they are $\boldsymbol{a}_1 = (0,0)$, $\boldsymbol{a}_2 = (0,1)$, $\boldsymbol{a}_3 = (1,0)$, $\boldsymbol{a}_4 = (1,1)$. Some noises are added in both problems with the noisy factor at 0.001. And the following root mean square distance is used to measure the accuracy of the locations of estimated sensors:

$$\text{RMSD} = \frac{1}{n}\sum_{i=1}^{n}\|\bar{\boldsymbol{x}}_i - \boldsymbol{x}_i^*\|_2^2,$$

where, $\bar{\boldsymbol{x}}_i$ and $\boldsymbol{x}_i^*$ are the estimated and true positions, respectively.

Experimental results are given in Tables 6 and 7, respectively. For the problem with 10 sensors and 4 anchors, the RMSD is 1.8943e−4, while for the problem with 20 sensors and 4 anchors, the RMSD is 9.6689e−5. The corresponding computed locations are illustrated in Figs. 3 and 4, respectively. We can find that the computed locations can coincide with the true locations.

Acknowledgments

References

[1] R.W. Cottle, Manifestations of the Schur complement, Linear Algebra and its Applications 8 (3) (1974) 189–211.
[2] I. Ekeland, R. Temam, Convex Analysis and Variational Problems, North-Holland, 1976.
[3] D.Y. Gao, Canonical dual transformation method and generalized triality theory in nonsmooth global optimization, Journal of Global Optimization 17 (2000) 127–160.
[4] D.Y. Gao, Duality Principles in Nonconvex Systems: Theory, Methods and Applications, Kluwer Academic, Dordrecht/Boston, 2000.
[5] D.Y. Gao, Canonical duality theory: unified understanding and generalized solution for global optimization problems, Computers & Chemistry 33 (2009) 1964–1972.
[6] D.Y. Gao, N. Ruan, P.M. Pardalos, Canonical dual solutions to sum of fourth-order polynomials minimization problems with applications to sensor network localization, Sensors: Theory, Algorithms and Applications 61 (1) (2012) 37–54.
[7] D.Y. Gao, G. Strang, Geometric nonlinearity: potential energy, complementary energy, and the gap function, Quarterly of Applied Mathematics 47 (1989) 487–504.
[8] D.Y. Gao, H.F. Yu, Multi-scale modelling and canonical dual finite element method in phase transitions of solids, International Journal of Solids and Structures 45 (13) (2008) 3660–3673.
[9] D. Jibetean, M. Laurent, Semidefinite approximations for global unconstrained polynomial optimization, SIAM Journal on Optimization 16 (2) (2005) 490–514.
[10] S.Y. Kim, M. Kojima, H. Waki, Generalized Lagrangian duals and sums of squares relaxations of sparse polynomial optimization problems, SIAM Journal on Optimization 15 (3) (2005) 697–719.
[11] S.Y. Kim, M. Kojima, H. Waki, M. Yamashita, Algorithm 920: SFSDP: a sparse version of full semidefinite programming relaxation for sensor network localization problems, ACM Transactions on Mathematical Software (TOMS) 38 (4) (2012) 27.
[12] J. Lasserre, B: Convergent SDP-relaxations in polynomial optimization with sparsity, SIAM Journal on Optimization 17 (3) (2006) 822–843.
[13] S. Pironio, M. Navascués, A. Acín, Convergent relaxations of polynomial optimization problems with noncommuting variables, SIAM Journal on Optimization 20 (5) (2010) 2157–2180.
[14] N. Ruan, D.Y. Gao, Canonical duality approach for nonlinear dynamical systems, IMA Journal of Applied Mathematics (2012), http://dx.doi.org/10.1093/imamat/hxs067.
[15] H.A.F.A. Santos, D.Y. Gao, Canonical dual finite element method for solving post-buckling problems of a large deformation elastic beam, International Journal of Nonlinear Mechanics 47 (2) (2011) 240–247.
[16] J.F. Sturn, Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, Optimization Methods and Software 11 (1999) 625–653.
[17] P. Tseng, Second-order cone programming relaxation of sensor network localization, SIAM Journal on Optimization 18 (1) (2007) 156–185.
[18] H. Waki, S.Y. Kim, M. Kojima, M. Muramatsu, Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity, SIAM Journal on Optimization 17 (1) (2006) 218–242.
[19] C. Wu, D.Y. Gao, K.L. Teo, A direct optimization method for low group delay FIR filter design, Signal Processing 93 (2013) 1764–1772.
[20] J. Zhang, D.Y. Gao, J. Yearwood, A novel canonical dual computational approach for prion AGAAAAGA amyloid fibril molecular modeling, Journal of Theoretical Biology 284 (1) (2011) 149–157.
[21] J.H. Zhu, X. Zhang, On global optimizations with polynomials, Optimization Letters 2 (2) (2008) 239–249.