POLITECNICO DI TORINO
Repository ISTITUZIONALE

Alternating direction method of multipliers for polynomial optimization

Original
Alternating direction method of multipliers for polynomial optimization / Cerone, V., Fosson, S., Pirrera, S., Regruto, D.. - ELETTRONICO. - (2023), pp. 1-6. (2023 European Control Conference (ECC) Bucharest, Romania 13-16 June 2023) [10.23919/ECC57647.2023.10178190].

Availability:
This version is available at: 11583/2980562 since: 2023-10-20T08:23:42Z

Publisher:
IEEE

Published
DOI:10.23919/ECC57647.2023.10178190

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

25 November 2026

Alternating direction method of multipliers for polynomial optimization

V. Cerone S. M. Fosson S. Pirrera D. Regruto

***Abstract*— Multivariate polynomial optimization is a prevalent model for a number of engineering problems. From a mathematical viewpoint, polynomial optimization is challenging because it is non-convex. The Lasserre's theory, based on semidefinite relaxations, provides an effective tool to overcome this issue and to achieve the global optimum. However, this approach can be computationally complex for medium and large scale problems. For this motivation, in this work, we investigate a local minimization approach, based on the alternating direction method of multipliers, which is low-complex, straightforward to implement, and prone to decentralization. The core of the work is the development of the algorithm tailored to polynomial optimization, along with the proof of its convergence. Through a numerical example we show a practical implementation and test the effectiveness of the proposed algorithm with respect to state-of-the-art methodologies.**

I. Introduction

Polynomial optimization problems (POPs) are concerned with the minimization of multivariate polynomial functions, over regions defined by polynomial equations and inequalities. Popular instances of POPs include linear, quadratic and mixed-integer programming, as well as partition problems on graphs, such as max-cut; see, e.g., [1] for a list of examples. Beyond that, a wide class of engineering problems can be formulated via POPs. Among them, we mention set-membership estimation and identification [2], [3], data-driven control [4], hybrid system identification [5], model predictive control [6], [7], and optimal control [8]. More recent applications include localization in narrowband internet-of-things [9], routing games [10], and estimation of Lipschitz constants in deep neural networks [11].

In general, POPs are non-convex and NP-hard, therefore the search of their global minima is challenging. To this purpose, solutions based on semidefinite programming (SDP) relaxations are proposed in the literature. In a nutshell, one can build a hierarchy of SDP problems that converge to the optimum under mild assumptions; see [12], [13], [14] for a complete overview. The SDP approach recast POPs into convex optimization and this makes the problem affordable. The development of the related theory is mature and several software implementations are available, such as GloptiPoly [15] and SparsePOP [16].

A drawback of the SDP approach is that the size of the involved matrices at d-degree of the hierarchy is proportional to $\begin{pmatrix} n+d \\ n \end{pmatrix}$, where n is the number of variables of the original POP. This becomes computationally prohibitive for medium and large scale models, in particular when high d are necessary to achieve the optimal solution. To tackle this problem, in the recent literature, SDP accelerating strategies are proposed, that exploit the presence of specific sparsity structures, such as the running intersection property [17], chordal sparsity [18], [19], or term sparsity [20]. However, these sparsity structures are not always present in SDP relaxations of POPs, as discussed, e.g., in [18]; therefore, these fast implementations are feasible only for some classes of POPs.

Given the described shortcomings of global optimization with SDP, one can relax POPs to the search of local minima or stationary points. In the literature, substantial attention is devoted to the local minimization of non-convex problems, e.g., via gradient-based methods, due also to the rise of deep learning techniques; we refer the reader to [21] for a recent survey. Convergence of the algorithms is a delicate point in this framework, both in terms of guarantees and speed.

In this work, we propose a novel approach to local minimization of POPs, based on the alternating direction method of multipliers (ADMM, [22]). ADMM is well-known in the context of convex optimization as an effective algorithm to minimize composite functionals, even in the presence of non-differentiable terms. ADMM structure is easy to implement and prone to decentralization. Its convergence is proven and analysed, see, e.g, [22], [23]. As to non-convex optimization, the convergence of ADMM is more critical to analyse and actually proven for some classes of problems; the main results are provided in [24], [25].

The implementation of ADMM for POPs is not straightforward, since POPs are non-convex, constrained problems, while ADMM is originally conceived for convex, unconstrained problems. The first contribution of this work is a suitable reformulation of POPs, on top of which we develop a novel ADMM-based procedure for POPs, that we name ADMM4POP. The second contribution is the proof of the convergence of ADMM4POP for the case of equality constraints, accounted for in a relaxed way. Furthermore, we illustrate the implementation of ADMM4POP in some numerical examples and we compare its practical effectiveness to state-of-the-art global/local minimization methods.

The paper is organized as follows. In Sec. II, we state the problem and we illustrate how to formulate it to apply ADMM. In Sec. III, we illustrate the details of the proposed ADMM4POP algorithm, while in Sec. IV we prove its convergence. In Sec. V, we propose two numerical examples. Finally, in Sec. VI we draw some conclusions.

* The authors are with the Dipartimento di Automatica e Informatica, Politecnico di Torino, corso Duca degli Abruzzi 24, 10129 Torino, Italy; e-mail: vito.cerone@polito.it, sophie.fosson@polito.it, simone.pirrera@polito.it, diego.regruto@polito.it;

II. Problem formulation

We consider a generic POP:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ & \text{s.t.} \\ & g_i(x) \leq 0 \quad i=1,\dots,l \\ & h_i(x) = 0 \quad i=1,\dots,m \end{aligned} \tag{1}$$

where f, g_i, h_i are multivariate polynomials of the decision variable $x \in \mathbb{R}^n$; see, e.g., [12] for details. The aim of this work is to develop an ADMM-based method for (1). For this purpose, first of all, we need to rewrite (1) as an unconstrained, composite problem, with some specific properties. In this section, we illustrate the proposed reformulation.

The first step is to write (1) as a quadratic optimization problem (QOP). As discussed, e.g., in [26], [27], any POP can be transformed into a QOP by adding suitable slack variables. Given a POP, different equivalent QOP representations are possible, as illustrated in [26, Sec. 2.3]. In the rest of the paper, given two vectors v, w of equal dimension, we write $v \leq w$ to denote the componentwise inequality. Moreover, we use the notation $A \succeq 0$ to indicate that a matrix A is positive semidefinite.

For our purpose, we consider any transformation that provides a QOP of this form:

$$\begin{aligned} \min_{x\in\mathbb{R}^{\tilde{n}}} \quad & \frac{1}{2}x^\top Ax + a^\top x \\ & \text{s.t.} \\ & Bx - b \leq 0 \\ & Cx - c = 0 \\ & x_i x_j = x_k, \quad (i,j,k) \in \mathcal{B} \end{aligned} \tag{2}$$

where $x \in \mathbb{R}^{\tilde{n}}$, $\tilde{n} \geq n$, is the augmented vector of decision variables; $A \in \mathbb{R}^{\tilde{n},\tilde{n}}$ is symmetric and $A \succeq 0$; $B \in \mathbb{R}^{l,\tilde{n}}$, $C \in \mathbb{R}^{\tilde{m},\tilde{n}}$, $a \in \mathbb{R}^{\tilde{n}}$, $b \in \mathbb{R}^l$, and $c \in \mathbb{R}^{\tilde{m}}$. We specify that $\tilde{m}$ is the number of equality constraints after the introduction of the slack variables. Moreover, $\mathcal{B} \subseteq \{1,\dots,\tilde{n}\}^3$ is the set of $3-$tuples (i,j,k) that contains all the indices such that $x_i x_j = x_k$.

Before illustrating the transformation to obtain (2), we provide some remarks.

Remark 1: If the transformation returns a QOP where A is not positive semidefinite, it is sufficient to introduce suitable quadratic constraints that reduce $\frac{1}{2}x^\top Ax + a^\top x$ to a linear function, i.e., $A = 0$.

Remark 2: Without loss of generality, we assume that for each $(i,j,k) \in \mathcal{B}$, the indices i, j and k are mutually different. If not, it is sufficient to add new slack variables. For example, in the presence of the constraint $x_i^2 = x_k$, i.e., $(i,i,k) \in \mathcal{B}$, we can define a new variable x_j and the constraint $x_j = x_i$ to obtain $x_i x_j = x_k$.

Remark 3: Without loss of generality, we assume that each index $i,j,k \in \{1,\dots,\tilde{n}\}$ appears at most once in each tuple $(i,j,k) \in \mathcal{B}$. This is not restrictive because if an overlap occurs, we can introduce an additional variable x_h, for each repeated variable x_i, together with constraints of the kind $x_h = x_i$.

For simplicity, we describe the transformation algorithm from (1) to (2) through an illustrative example.

Example 1: Let us consider the POP

$$\min_{x\in\mathbb{R}^2} x_1^2 x_2.$$

By defining $x_3 = x_1^2$, we obtain $\min_{x\in\mathbb{R}^2} x_3 x_2$ s.t. $x_1^2 = x_3$. Following Remark 1, we notice that $A = \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$ is not positive semi-definite. Therefore, we define $x_4 = x_3 x_2$.

Moreover, according to Remark 2, we define $x_5 = x_1$ to transform the constraint $x_1^2 = x_3$ into $x_1 x_5 = x_3$. In this way, $\mathcal{B} = \{(3,2,4),(1,5,3)\}$. To fulfill the non-overlap assumption described in Remark 3, we add $x_6 = x_3$. In conclusion, we obtain the QOP

$$\begin{aligned} \min_{x\in\mathbb{R}^5} \quad & x_4 \\ & \text{s.t.} \\ & x_5 = x_1, \qquad x_6 = x_3 \\ & x_i x_j = x_k, \ (i,j,k) \in \mathcal{B} = \{(3,2,4),(1,5,6)\}. \end{aligned}$$

Let us define the sets

$$\mathcal{P} \doteq \{x \in \mathbb{R}^{\tilde{n}} : x_i x_j = x_k, \quad \forall (i,j,k) \in \mathcal{B}\}. \tag{3}$$

and

$$\mathcal{D} \doteq \{x \in \mathbb{R}^{\tilde{n}} | Bx - b \leq 0, Cx - c = 0\}. \tag{4}$$

We notice that $\mathcal{P}$ is a non-convex semialgebraic set that describes all the non-convexity of the original problem. Instead, $\mathcal{D}$ is a convex set that includes all the linear constraints of (2). Given these definitions, we rewrite (2) as

$$\min_{x\in\mathcal{D}} \quad \frac{1}{2}x^\top Ax + a^\top x + \mathbb{1}_{\mathcal{P}}(x) \tag{5}$$

where $\mathbb{1}_{\mathcal{P}}(x)$ denotes the indicator function of $\mathcal{P}$, i.e., $\mathbb{1}_{\mathcal{P}}(x) = 0$ if $x \in \mathcal{P}$, and ∞ otherwise.

III. Proposed algorithm: ADMM4POP

In the previous section, we have shown that a generic POP can be rewritten as a QOP of the form (5). In this section, we prove that a POP formulated as in (5) is prone to be solved through ADMM.

In general, ADMM can be applied to composite problems of kind $\min_{x\in\mathbb{R}^n} F(x) + G(x)$; see [22] for details and more general formulations. More specifically, to apply ADMM we split the problem as follows:

$$\begin{aligned} \min_{x,z\in\mathbb{R}^n} \quad & F(x) + G(z) \\ & \text{s.t.} \qquad x = z \end{aligned} \tag{6}$$

and we consider the augmented Lagrangian

$$L_\rho(x,z) = F(x) + G(z) + \rho u^\top (x - z) + \frac{\rho}{2}\|x - z\|_2^2 \tag{7}$$

where $\rho > 0$ is a scalar penalty parameter and $u \in \mathbb{R}^n$ is the

scaled dual variable; we refer the reader to [22, Sec. 3.1.1.] for details on the scaled ADMM. In the following, we denote by x_k, z_k and u_k the estimates of the variables x, z and u, respectively, at a generic k-th iteration of the algorithm.

ADMM consists in a loop iterating over three steps:

1) x-update:

$$\begin{aligned} x_{\tau+1} &= \arg\min_{x\in\mathbb{R}^n} L_\rho(x, z) \\ &= \arg\min_{x\in\mathbb{R}^n} F(x) + \frac{\rho}{2}\|x - z_\tau + u_\tau\|_2^2. \end{aligned} \tag{8a}$$

2) z-update:

$$\begin{aligned} z_{\tau+1} &= \arg\min_{z\in\mathbb{R}^n} L_\rho(x, z) \\ &= \arg\min_{z\in\mathbb{R}^n} G(z) + \frac{\rho}{2}\|x_{\tau+1} - z + u_\tau\|_2^2. \end{aligned} \tag{8b}$$

3) u-update:

$$u_{\tau+1} = u_\tau + x_{\tau+1} - z_{\tau+1}. \tag{8c}$$

Our aim is to apply ADMM to (5). To this purpose, we decompose (5) as in (6), by setting $F(x) = \frac{1}{2}x^\top Ax + a^\top x$ and $G(z) = \mathbb{1}_{\mathcal{P}}(z)$. Then, we have

$$\begin{aligned} \min_{x\in\mathcal{D}, z\in\mathbb{R}^{\tilde{n}}} \quad & \frac{1}{2}x^\top Ax + a^\top x + \mathbb{1}_{\mathcal{P}}(z) \\ \text{s.t.} \quad & x = z \end{aligned} \tag{9}$$

and we implement the algorithm as in (8a)-(8b)-(8c). While step (8c) is trivial, (8a) and (8b) require more computations.

A. x-update step

According to (8a), the x-update applied on (9) is

$$x_{\tau+1} = \arg\min_{x\in\mathcal{D}} \frac{1}{2}x^\top Ax + a^\top x + \frac{\rho}{2}\|x - z_\tau + u_\tau\|_2^2. \tag{10}$$

This is equivalent to

$$\begin{aligned} x_{\tau+1} = \arg\min_{x\in\mathbb{R}^{\tilde{n}}} \quad & \frac{1}{2}x^\top \left(A + \rho I_{\tilde{n}}\right) x + x^\top \left[a + \rho(u_\tau - z_\tau)\right] \\ \text{s.t.} \quad & \\ & Bx - b \leq 0, \\ & Cx - c = 0 \end{aligned} \tag{11}$$

where $I_{\tilde{n}}$ is the identity matrix of dimension $\tilde{n}$. This is a convex quadratic program (QP) and can be solved efficiently to global optimality.

Moreover, if no inequality constraints are present, the solution of (11) reduces to the solution of the linear system of the Karush-Kuhn-Tucker (KKT) conditions:

$$\begin{bmatrix} A + \rho I & C^\top \\ C & 0 \end{bmatrix} \begin{bmatrix} x_{\tau+1} \\ \mu \end{bmatrix} = \begin{bmatrix} \rho(z_\tau - u_\tau) - a \\ c \end{bmatrix} \tag{12}$$

where μ is the Lagrange multiplier; see [28] for details. In sections IV and V we also discuss the relaxation of the equality constraints.

B. z-update step

According to (8b), the z-update applied on (9) is

$$z_{\tau+1} = \arg\min_{z\in\mathbb{R}^{\tilde{n}}} \mathbb{1}_{\mathcal{P}}(z) + \frac{\rho}{2}\|x_{\tau+1} - z + u_\tau\|_2^2. \tag{13}$$

This is equivalent to

$$z_{\tau+1} = \arg\min_{z\in\mathcal{P}} \|z - (x_{\tau+1} + u_\tau)\|_2^2. \tag{14}$$

Since $\mathcal{P}$ is non-convex, the solution to (14) is not straightforward. However, we can exploit the decoupling assumption discussed in Remark 3 to split the problem into smaller and affordable sub-problems. More precisely, on the one hand the quadratic cost in (14) is separable in its components; on the other hand, according to the decoupling assumption, we can partition $\mathcal{P}$ into separated subsets, each of them representing a constraint $x_i x_j = x_k$. Let $t = 1, \ldots, |\mathcal{B}|$ be the ordered indices of the elements of $\mathcal{B}$, i.e., each $(i, j, k) \in \mathcal{B}$ is labeled with a $t \in \{1, \ldots, |\mathcal{B}|\}$. Then, we can write

$$\mathcal{P} = \mathcal{P}_1 \times \mathcal{P}_2 \times \cdots \times \mathcal{P}_{|\mathcal{B}|} \tag{15}$$

where

$$\mathcal{P}_t \doteq \{(x_i, x_j, x_k) : x_i x_j = x_k\} \text{ for each } t = 1, \ldots, |\mathcal{B}|. \tag{16}$$

Hence,

$$z_{\tau+1} = \arg\min_{z\in\mathcal{P}_1\times\cdots\times\mathcal{P}_{|\mathcal{B}|}} \sum_{t=1}^{|\mathcal{B}|} \|z_t - (x_{\tau+1} + u_\tau)_t\|_2^2 \tag{17}$$

where $w_t \doteq (w_i, w_j, w_k)$ for any vector $w \in \mathbb{R}^{\tilde{n}}$ and (i, j, k) corresponding to the index t.

In conclusion, (17) can be split into $|\mathcal{B}|$ sub-problems with only three variables, i.e.,

$$(z_{\tau+1})_t = \arg\min_{z_t\in\mathcal{P}_t} \|z_t - (x_{\tau+1} + u_\tau)_t\|_2^2. \tag{18}$$

In other terms, $(z_{\tau+1})_t$ is the ℓ_2 projection of $v_t = (v_i, v_j, v_k) \doteq (x_{\tau+1} + u_\tau)_t \in \mathbb{R}^3$ onto $\mathcal{P}_t$. The problems in (18) are non-convex; however, we can solve them in closed form. In fact, we have

$$\begin{aligned} (z_{\tau+1})_t = \arg\min_{z_t\in\mathbb{R}^3} \quad & (z_i - v_i)^2 + (z_j - v_j)^2 + (z_k - v_k)^2 \\ \text{s.t.} \quad & z_i z_j = z_k. \end{aligned} \tag{19}$$

Then, we plug $z_k = z_i z_j$ into the cost functional, and remove the corresponding constraint:

$$(z_{\tau+1})_t = \arg\min_{z_t\in\mathbb{R}^3} (z_i - v_i)^2 + (z_j - v_j)^2 + (z_i z_j - v_k)^2. \tag{20}$$

Now, we can explicitly find the minima by evaluating the first order conditions: given $f_v(z_i, z_j) = (z_i - v_i)^2 + (z_j - v_j)^2 + (z_i z_j - v_k)^2$, we solve

$$\nabla f_v(z_i, z_j) = 2\begin{pmatrix} z_i - v_i + z_i z_j^2 - z_j v_k \\ z_j - v_j + z_i^2 z_j - z_i v_k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{21}$$

From the first equation, we obtain

$$z_i - v_i + z_i z_j^2 - z_j v_k = 0 \Rightarrow z_i = \frac{v_i + z_j v_k}{1 + z_j^2} \tag{22}$$

By replacing (22) into the second equation of (21), we get

$$\begin{aligned} z_j - v_j + \left(\frac{v_i + z_j v_k}{1 + z_j^2}\right)^2 z_j - \frac{v_i + z_j v_k}{1 + z_j^2} v_k = \\ = \frac{z_j^5 + c_4 z_j^4 + c_3 z_j^3 + c_2 z_j^2 + c_1 z_j + c_0}{\left(1 + z_j^2\right)^2} = 0 \end{aligned} \tag{23}$$

where

$$\begin{aligned} c_0 = -v_j - v_i v_k, \quad c_1 = v_i^2 - v_k^2 + 1, \\ c_2 = v_i v_k - 2 v_j, \quad c_3 = 2, \quad c_4 = -v_j. \end{aligned} \tag{24}$$

As $1 + z_j^2 \neq 0$, we obtain the candidate solutions by finding the zeros of a univariate polynomial of degree 5, e.g., by computing the eigenvalues of the companion matrix

$$\left[\begin{array}{c|c} 0 & I_4 \\ \hline -c_0 & -c_1 \ \ldots \ -c_4 \end{array}\right]. \tag{25}$$

In this way, we get five candidate solutions, which are either real or complex and conjugate, and at least one of them is real. In turn, we evaluate the cost functional on each real candidate solution, which finally provides the desired minimum.

We notice that the z-update requires to compute the eigenvalues of a 5×5 matrix for each bilinear constraint, which may be burdensome if $|\mathcal{B}|$ is large. However, the problem can be fully parallelized, by exploiting the separability of $\mathcal{B}$; therefore, in the presence of suitable hardware, the z-update can be solved very effectively.

Algorithm 1 ADMM4POP

1: Initialization: $z_0, u_0 \in \mathbb{R}^{\tilde{n}}$
2: **for all** $\tau = 1, \ldots, T_{stop}$ **do**
3: $x_{\tau+1} = \arg\min_{x \in \mathcal{D}} \frac{1}{2} x^\top A x + a^\top x + \frac{\rho}{2} \|x - z_\tau + u_\tau\|_2^2$, via convex QP or KKT conditions, see Sec. III-A
4: For each $t = 1, \ldots, |\mathcal{B}|$, $(z_{\tau+1})_t = \arg\min_{z_t \in \mathbb{R}^3} (z_i - v_i)^2 + (z_j - v_j)^2 + (z_i z_j - v_k)^2$ by explicit computation, see Sec. III-B
5: $u_{\tau+1} = u_\tau + x_{\tau+1} - z_{\tau+1}$
6: **end for**

We summarize the proposed algorithm, denoted as ADMM4POP, in Algorithm 1.

Concerning the stopping time T_{stop}, as suggested in [22], we stop the algorithm when the norms of both primal and dual residuals $\|x_\tau - z_\tau\|$ and $\|\rho(z_{\tau-1} - z_\tau)\|$ are sufficiently small.

IV. Convergence analysis

Proving the convergence of ADMM in non-convex problems is a challenging task. In [29], the authors prove the convergence to a stationary point of single-block ADMM for a family of non-convex problems. Later, in [24], [25] extensions to multi-block problems are proposed. In [24], the terms of the cost functional that are non-convex must be differentiable to prove the convergence. This is not the case of (5), where the indicator function of a non-convex set occurs, which is non-convex and non-smooth. Since in [29] the authors account for terms that are both non-convex and non-smooth, we leverage their results to develop the convergence analysis of ADMM4POP.

We remark that the analysis in [29] is developed for unconstrained problems; thus, for simplicity, in this work we recast problem (2) into unconstrained optimization as well, while we leave the proof of convergence of the constrained version for future extended work. To this end, in (2), we assume that there are no inequality constraints $Bx \leq b$, while the equality constraints $Cx = c$ are accounted for in a relaxed way, i.e., by adding the term $\gamma \|Cx - c\|_2^2$ to the objective function, for some $\gamma > 0$. Finally, we keep the bilinear constraints, because we represent them through the indicator function, see (5). In conclusion, we consider

$$\min_{x \in \mathbb{R}^{\tilde{n}}} \quad \frac{1}{2} x^\top A x + a^\top x + \gamma \|Cx - c\|_2^2 + \mathbb{1}_P(x). \tag{26}$$

As $\gamma \|Cx - c\|_2^2$ is quadratic and convex, (26) has the form of (5); then, we can apply ADMM4POP to it. We refer to this approach as relaxed ADMM4POP.

For completeness, we summarize a result from [29], which is the basis for our proof of convergence. Let us consider the composite problem (6) and its augmented Lagrangian (7), under the following conditions.

Assumption 1:

A1. F is semi-algebraic, twice continuously differentiable on $\mathbb{R}^n$ with a bounded, positive semidefinite Hessian $\nabla^2 F$;
A2. G is a semi-algebraic, proper, lower semicontinuous function;
A3. there exists $0 < \zeta < \rho$ such that $F(x) - \frac{1}{2\zeta} \|\nabla F(x)\|^2$ is lower bounded;
A4. the design parameter ρ is designed so that $\rho I_n - \sqrt{2} \nabla^2 F(x) \succeq 0$.

We recall that a function is proper if it is finite somewhere and it does tend to $-\infty$; see, e.g., [29, Sec. 2].

Theorem 1: [29, Theorems 1-2-4] Let us suppose that Assumption 1 holds. Then, the sequence (x_k, z_k, u_k) generated by ADMM converges to a point $(x^\star, z^\star, u^\star)$, and $x^\star$ is a stationary point of $F(x) + G(x)$.

Proposition 1: Let us consider (26). The sequence (x_k, z_k, u_k) generated by relaxed ADMM4POP converges to a point $(x^\star, z^\star, u^\star)$, and $x^\star$ is a stationary point of (26).

Proof: Let us consider $F(x) = \frac{1}{2} x^\top A x + a^\top x + \gamma \|Cx - c\|_2^2$ and $G(z) = \mathbb{1}_P(z)$. In the following, we prove that all the points of Assumption 1 are fulfilled, which is sufficient to prove the thesis.

By construction of A, it is straightforward to prove that $\frac{1}{2} x^\top A x + a^\top x + \gamma \|Cx - c\|_2^2$ fulfills A1.

Regarding A2, by definition, $\mathbb{1}_P$ is proper. Moreover, it is

semi-algebraic, because $\mathcal{P}$ is a semi-algebraic set, and the indicator function of a semi-algebraic set is semi-algebraic, see, e.g., [30]. Finally, the indicator function of a set is lower semi-continuous if the set is closed; then, we need to show that $\mathcal{P}$ is closed. For this purpose, let t be the index of (i,j,k) as defined in Sec. II and let us define

$$\mathcal{Q}_t \doteq \left\{ z \in \mathbb{R}^{\tilde{n}} : z_i z_j = z_k \right\}. \tag{27}$$

We remark that $\mathcal{Q}_t$ is different from $\mathcal{P}_t$ defined in (15). In particular, it holds $\mathcal{P} = \bigcup_{t\in\mathcal{B}} \mathcal{Q}_t$. Since $\mathcal{Q}_t = \{z : -z_i z_j + z_k \leq 0\} \cup \{z : z_i z_j - z_k \leq 0\}$ and since $z_i z_j - z_k$ and $-z_i z_j + z_k$ are continuous functions, their sub-level sets are closed. As a consequence $\mathcal{Q}_t$ is an intersection of closed sets, hence it is closed. In turn, $\mathcal{P}$ is a union of closed sets, hence it is closed. Then, we conclude that $\mathbb{1}_{\mathcal{P}}$ is lower semi-continuous.

Finally, A3 holds because, in our problem, $F(x) - \frac{1}{2\zeta}\|\nabla F(x)\|^2$ is convex if we suitably design $\rho > \zeta > \|A + \gamma C^\top C\|_2$. In particular, if we set $\rho > \sqrt{2}\|A+\gamma C^\top C\|_2$, A4 is satisfied as well. ■

V. Numerical experiments

In this section we propose two numerical experiments to test the effectiveness of the proposed ADMM4POP and to validate the theoretical convergence results. For ADMM4POP, we implement both the constrained version and the relaxed one, which is guaranteed to converge by Proposition 1. However, the constrained version as well is convergent in our experiments.

We compare ADMM4POP to two state-of-the-art methods, namely the interior-point method (IPM) and the SDP approach [12]. IPM is a local algorithm like ADMM4POP. All the experiments are performed with MATLAB R2021b, on a PC with AMD Ryzen 7 1700 8-Core CPU @ 3.00 GHz and 16 GB RAM. We implement IPM through the function *fmincon* in the MATLAB Optimization Toolbox, while we use the SparsePOP package [16] to implement the SDP method.

In the first experiment, we consider the following POP:

$$\begin{aligned} \min_{x\in\mathbb{R}^3} \quad & x_1^2 x_2^2 + x_1^2 + q_1 x_1 + x_2^2 + x_2 x_3 + q_2 x_2 + x_3^2 + q_3 x_3 \\ & \text{s.t.} \\ & x_2 x_3 + x_1 = 10 \end{aligned}$$

where $q_1 \in [4,6]$, $q_2 \in [-8,-6]$, and $q_3 \in [1,3]$ are generated uniformly at random. We reformulate the problem as in (2). First, we replace $x_4 = x_2 x_3$ and $x_5 = x_1 x_2$. Then, we define $x_6 = x_2$ to fulfill the splitting condition of Remark 3. In conclusion, we obtain

$$\begin{aligned} \min_{x\in\mathbb{R}^6} \quad & x_5^2 + x_1^2 + q_1 x_1 + x_2^2 + x_4 + q_2 x_2 + x_3^2 + q_3 x_3 \\ & \text{s.t.} \\ & x_4 + x_1 = 10, \qquad x_6 = x_2 \\ & x_i x_j = x_k, \ (i,j,k) \in \mathcal{B} = \{(2,3,4),(1,6,5)\} \end{aligned}$$

which is in the form (5). In particular, it is straightforward to verify that the quadratic cost functional is convex.

The stop threshold for ADMM4POP is set so that the obtained accuracy is comparable to that one of IPM. We randomly initialize all the variables with standard Gaussian distribution. We set $\rho = 2$, and $\gamma = 10^3$ for relaxed ADMM4POP. In this experiment, SDP with relaxation order 2 always achieves the global minimum, thus it is used as reference.

In Table I, we report the results over 500 random runs, in terms of accuracy and runtime statistics. We evaluate the error as the ℓ_2 distance from the SDP solution, which is considered as ground truth. Both ADMM4POP and IPM substantially achieve the desired global minimum. As expected, in the relaxed version of ADMM4POP the error is slightly larger; however, the gap is not relevant in practice, that is, the global minimizer is identified with sufficient accuracy. As to the runtime, we can see that ADMM4POP is significantly faster than IPM and SDP.

	Error	Runtime (ms)		
	mean	mean	min	max
SDP	–	63.5	59.0	88.5
IPM	2.5×10^{-5}	12.3	7.7	21.8
ADMM4POP cns	6.5×10^{-5}	1.6	1.5	7.9
ADMM4POP rel	4.2×10^{-4}	1.7	1.6	8.9

TABLE I

First experiment: constrained POP; statistics over 500 random runs. ADMM4POP is implemented in both constrained (cns) and relaxed (rel) versions.

In the second experiment, we deal with a system identification problem. We consider the discrete-time dynamic system $y(k+1) = \alpha y(k) + \beta u(k+1)$, $k = 0,\dots,K$, and we aim at recovering the parameters α and β given the knowledge of the input $u \in \mathbb{R}^K$ and noisy measurements of the output $w \doteq y + \eta$, where $\eta \in \mathbb{R}^K$ represents the noise. In particular, we minimize the simulation error ℓ_2-norm, i.e. we solve

$$\begin{aligned} \min_{y\in\mathbb{R}^K} \quad & \|y - w\|_2^2 \\ & \text{s.t.} \\ & y(k+1) = \alpha y(k) + \beta u(k+1) \quad k = 1,\dots,K-1. \end{aligned}$$

As illustrated in Sec. II, we define the slack variables $\psi_k = \alpha$ and $\xi_k = \psi_k y(k)$ for $k = 1,\dots,K-1$, to decouple the bilinear constraints. Thus, $\theta = (\alpha, \beta, y(1),\dots,y(K), \psi_1,\dots,\psi_{K-1}, \xi_1,\dots,\xi_{K-1}) \in \mathbb{R}^{2+K+2(K-1)}$ is the total vector of variables, and $\mathcal{B} = \{(2+h, 2+K+h, 1+2K+h), h = 1,\dots,K-1\}$.

We perform 500 random runs, with $\alpha, \beta \in (-1,-0.5) \cup (0.5,1)$ generated uniformly at random; η is a white Gaussian noise with variance 10^{-4}. The variables are initialized with Gaussian distribution. In each run, we measure $K = 500$ output samples, the total number of variables being significantly larger with respect to the first experiment. For ADMM4POP, we set $\rho = 1$ and $\gamma = 10$. For the SDP approach, we set the relaxation order to 2, which is observed to be sufficient to achieve the global minimum.

	Error	Runtime (s)		
	Mean	mean	min	max
SDP	6.1×10^{-4}	10.16	5.87	22.99
IPM	3.2×10^{-4}	3.97	1.34	36.73
ADMM4POP cns	3.6×10^{-4}	2.53	0.76	21.45
ADMM4POP rel	3.9×10^{-4}	0.62	0.40	1.61

TABLE II

SECOND EXPERIMENT: SYSTEM IDENTIFICATION; STATISTICS OVER 500 RANDOM RUNS. ADMM4POP IS IMPLEMENTED IN BOTH CONSTRAINED (CNS) AND RELAXED (REL) VERSIONS.

In Table II, we collect the results. We define the error as the ℓ_2 distance between the estimates and the true parameters α, β. All the considered algorithms identify the correct parameters, with similar accuracy. Regarding the runtime, ADMM4POP is faster than the other algorithms, with a more evident improvement in the relaxed version.

We remark that, in the proposed experiments, we perform the decoupled projections (18) sequentially, which leaves room for further enhancement via parallelization, in particular in view of larger dimension problems.

VI. CONCLUSIONS

In this work, we develop ADMM4POP, a low-complex strategy to implement the alternating direction method of multipliers for local minimization of polynomial optimization problems. By developing a suitable formulation of the problem, we prove that we can easily implement ADMM4POP for polynomial optimization, both in constrained and relaxed versions. Moreover, we prove its convergence to a stationary point in the relaxed versions. Through numerical examples, we show that ADMM4POP achieves the global minimum of polynomial problems with reduced run time with respect to state-of-the-art methods. As ADMM4POP is prone to decentralization, future work will envisage the parallelization of the proposed method and its application to large scale polynomial problems.

REFERENCES

[1] M. Laurent, *Sums of Squares, Moment Matrices and Optimization Over Polynomials*. Springer New York, 2009, pp. 157–270.

[2] V. Cerone, D. Piga, and D. Regruto, "Set-membership error-in-variables identification through convex relaxation techniques," *IEEE Trans. Autom. Control*, vol. 57, no. 2, pp. 517–522, 2012.

[3] V. Cerone, J.-B. Lasserre, D. Piga, and D. Regruto, "A unified framework for solving a general class of conditional and robust set-membership estimation problems," *IEEE Trans. Autom. Control*, vol. 59, no. 11, pp. 2897–2909, 2014.

[4] V. Cerone, D. Regruto, and M. Abuabiah, "Direct data-driven control design through set-membership errors-in-variables identification techniques," in *Proc. Amer. Control Conf. (ACC)*, 2017, pp. 388–393.

[5] C. Feng, C. M. Lagoa, and M. Sznaier, "Hybrid system identification via sparse polynomial optimization," in *Proc. Amer. Control Conf. (ACC)*, 2010, pp. 160–165.

[6] M. Canale, V. Cerone, D. Piga, and D. Regruto, "Approximation of model predictive control laws for polynomial systems," *Asian J. Control*, vol. 16, no. 5, pp. 1425–1436, 2014.

[7] E. Harinath, L. C. Foguth, J. A. Paulson, and R. D. Braatz, "Nonlinear model predictive control using polynomial optimization methods," in *Proc. Amer. Control Conf. (ACC)*, 2016, pp. 1–6.

[8] D. Rodrigues and A. Mesbah, "Efficient global solutions to single-input optimal control problems via approximation by sum-of-squares polynomials," *IEEE Trans. Autom. Control*, vol. 67, no. 9, pp. 4674–4686, 2022.

[9] S. Sedighi, K. V. Mishra, M. R. B. Shankar, and B. Ottersten, "Localization with one-bit passive radars in narrowband internet-of-things using multivariate polynomial optimization," *IEEE Trans. Signal Process.*, vol. 69, pp. 2525–2540, 2021.

[10] Y. Zhu and K. Savla, "Information design in nonatomic routing games with partial participation: Computation and properties," *IEEE Trans. Control Netw. Syst.*, vol. 9, no. 2, pp. 613–624, 2022.

[11] T. Chen, J. Lasserre, V. Magron, and E. Pauwels, "Semialgebraic optimization for Lipschitz constants of ReLU networks," in *Adv. Neural Inform. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 19 189–19 200.

[12] J.-B. Lasserre, *An introduction to polynomial and semi-algebraic optimization*. Cambridge Univ. Press, 2015.

[13] P. A. Parrilo, "Semidefinite programming relaxations for semialgebraic problems," *Math. Program.*, vol. 96, no. 2, pp. 293–320, 2003.

[14] N.-Z. Shor, "Quadratic optimization problems," *Sov. J. Comput. Syst. Sci*, vol. 26, 1987.

[15] D. Henrion and J.-B. Lasserre, "GloptiPoly: global optimization over polynomials with Matlab and SeDuMi," in *Proc. IEEE Conf. Decis. Control (CDC)*, vol. 1, 2002, pp. 747–752 vol.1.

[16] H. Waki, S. Kim, M. Kojima, M. Muramatsu, and H. Sugimoto, "SparsePOP: a sparse semidefinite programming relaxation of polynomial optimization problems," *ACM Trans. Math. Softw.*, vol. 35, 2008.

[17] T. Weisser, J. B. Lasserre, and K.-C. Toh, "Sparse-BSOS: a bounded degree SOS hierarchy for large scale polynomial optimization with sparsity," *Math. Programm. Comput.*, vol. 10, pp. 1–32, 2018.

[18] Y. Zheng, G. Fantuzzi, A. Papachristodoulou, P. Goulart, and A. Wynn, "Chordal decomposition in operator-splitting methods for sparse semidefinite programs," *Math. Progamm.*, 2019.

[19] Y. Zheng, G. Fantuzzi, and A. Papachristodoulou, "Fast ADMM for Sum-of-Squares Programs Using Partial Orthogonality," *IEEE Trans. Autom. Control*, vol. 64, no. 9, pp. 3869–3876, 2019.

[20] J. Wang, V. Magron, and J.-B. Lasserre, "TSSOS: A Moment-SOS Hierarchy That Exploits Term Sparsity," *SIAM J. Optim.*, vol. 31, no. 1, pp. 30–58, 2021.

[21] M. Danilova, P. Dvurechensky, A. Gasnikov, E. Gorbunov, S. Guminov, D. Kamzolov, and I. Shibaev, *Recent Theoretical Advances in Non-Convex Optimization*. Springer Int., 2022, pp. 79–163.

[22] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. Now Foundations and Trends, 2011.

[23] M. Hong and Z. Luo, "On the linear convergence of the alternating direction method of multipliers," *Math. Programm.*, vol. 162, pp. 165–199, 2017.

[24] M. Hong, Z. Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM J. Optim.*, vol. 26, no. 1, pp. 337–364, 2016.

[25] Y. Wang, W. Yin, and J. Zeng, "Global convergence of ADMM in nonconvex nonsmooth optimization," *J. Sci. Comput.*, vol. 78, no. 1, pp. 29–63, jan 2019.

[26] M. Mevissen and M. Kojima, "SDP relaxations for quadratic optimization problems derived from polynomial optimization problems," *Asia Pac. J. Oper. Res.*, vol. 27, pp. 15–38, 02 2010.

[27] S. Elloumi, A. Lambert, and A. Lazare, "Semidefinite programming relaxations through quadratic reformulation for box-constrained polynomial optimization problems," in *Int. Conf. Control Decis. Inf. Technol. (CoDIT)*. IEEE, 2019, pp. 1498–1503.

[28] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[29] G. Li and T. K. Pong, "Global convergence of splitting methods for nonconvex composite optimization," *SIAM J. Optim.*, vol. 25, no. 4, pp. 2434–2460, 2015.

[30] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, "Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality," *Math. Oper. Res.*, vol. 35, no. 2, pp. 438–457, 2010.