Rank-2 Matrix Solution for Semidefinite Relaxations of Arbitrary Polynomial Optimization Problems

Ramtin Madani, Ghazal Fazelnia and Javad Lavaei

***Abstract*—This paper is concerned with the study of an arbitrary polynomial optimization via a convex relaxation, namely a semidefinite program (SDP). The existence of a rank-1 matrix solution for the SDP relaxation guarantees the recovery of a global solution of the original problem. The main objective of this work is to show that an arbitrary polynomial optimization has an equivalent formulation in the form of a sparse quadratically-constrained quadratic program (QCQP) whose SDP relaxation possesses a matrix solution with rank at most 2. This result offers two new insights into the computational complexity of polynomial optimization and combinatorial optimization as a special case. First, the complexity is only related to finding a rank-1 matrix in a convex set where it is guaranteed that a rank-2 matrix can always be found in polynomial time. Second, the approximation of the rank-2 SDP solution with a rank-1 matrix enables the retrieval of an approximate near-global solution for the original polynomial optimization. To derive this result, three graph sparsification techniques are proposed, each of which designs a sparse QCQP problem that is equivalent to the original polynomial optimization.**

I. Introduction

Optimization theory deals with the minimization of an objective function subject to a set of constraints. This area plays a vital role in the design, control, operation, and analysis of real-world systems. The development of efficient optimization techniques and numerical algorithms has been an active area of research for many decades. The goal is to design a robust, scalable method that is able to find a global solution in polynomial time. This has been fully answered for the class of convex optimization problems that includes all linear and some nonlinear problems [1]–[3]. Convex optimization has found a wide range of applications across engineering and economics [4]. In the past several years, a great effort has been devoted to casting real-world problems as convex optimization. Nevertheless, several classes of optimization problems, including polynomial optimization and quadratically constrained quadratic program (QCQP) as a special case, are nonlinear, non-convex, and NP-hard in the worst case [5], [6]. In particular, there is no known effective optimization technique for integer and combinatorial optimization as a small subclass of QCQP [7], [8]. Given a non-convex optimization, there are several techniques to find a solution that is locally optimal. However, seeking a global or near global solution in polynomial time is a daunting challenge. There is a large body of literature on nonlinear optimization witnessing the complexity of this problem.

The authors are with the Electrical Engineering Department, Columbia University (emails: madani@ee.columbia.edu, gf2293@columbia.edu, and lavaei@ee.columbia.edu). This work was supported by a Google Research Award.

To reduce the computational complexity of a non-convex optimization, several convex relaxation methods based on linear matrix inequality (LMI), semidefinite programming (SDP), and second-order cone programming (SOCP) have gained popularity [1]–[9]. These techniques enlarge the possibly non-convex feasible set into a convex set characterizable via convex functions, and then provide the exact or a lower bound on the optimal objective value associated with a global solution. The effectiveness of this technique has been substantiated in different contexts [10]–[20]. The SDP relaxation converts an optimization with a vector variable to a convex optimization with a matrix variable, via a lifting technique. The exactness of the relaxation can then be interpreted as the existence of a low-rank (e.g., rank-1) matrix solution for the SDP relaxation. Several papers have studied the existence of a low-rank solution to matrix optimizations with linear and LMI constraints [21]–[25].

In this paper, we aim to prove that an arbitrary polynomial optimization can be equivalently converted to a sparse QCQP whose SDP relaxation possesses a matrix solution with rank at most 2. Since the existence of a rank-1 solution guarantees the recovery of a global solution of the original problem, this result implies that the computational complexity of a general polynomial optimization is only related to the hardness of transforming an SDP solution with rank at most 2 to an optimal rank-1 matrix. To elaborate on this result, consider a polynomial optimization

$$\min_{\mathbf{x}\in\mathbb{R}^q} \quad P_0(\mathbf{x}) \tag{1a}$$
$$\text{s.t.} \quad P_k(\mathbf{x}) \leq 0 \quad \text{for} \quad k=1,2,\ldots,l \tag{1b}$$

where $P_0(\mathbf{x}),\ldots,P_l(\mathbf{x})$ are arbitrary polynomial functions. Assume that this optimization has a feasible solution. The above problem can be reformulated as the QCQP

$$\min_{\mathbf{u}\in\mathbb{R}^n} \quad \mathbf{u}^T\mathbf{M}_0\mathbf{u} \tag{2a}$$
$$\text{s.t.} \quad \mathbf{u}^T\mathbf{M}_k\mathbf{u} \leq y_k \quad \text{for} \quad k=1,2,\ldots,r \tag{2b}$$

for some (fixed) non-unique numbers n, r, and $y_1,y_2,\ldots,y_r$. In this problem, the variable $\mathbf{u}$ consists of multiple copies of the entries of $\mathbf{x}$ as well as some auxiliary parameters, and the matrices $\mathbf{M}_0,\mathbf{M}_1,\ldots,\mathbf{M}_r$ are all sparse. The above QCQP can be cast as

$$\min_{\mathbf{W}\in\mathbb{S}^n} \quad \text{trace}\{\mathbf{M}_0\mathbf{W}\} \tag{3a}$$
$$\text{s.t.} \quad \text{trace}\{\mathbf{M}_k\mathbf{W}\} \leq y_k \quad \text{for} \quad k=1,2,\ldots,r \tag{3b}$$
$$\mathbf{W} \succeq 0 \tag{3c}$$
$$\text{rank}\{\mathbf{W}\} = 1 \tag{3d}$$

where the positive semidefinite matrix $\mathbf{W}$ plays the role of

$\mathbf{u}^T\mathbf{u}$. The SDP relaxation of the QCQP problem (2) can be obtained by dropping the rank constraint from the above non-convex optimization, which results in the convex program

$$\min_{\mathbf{W}\in\mathbb{S}^n} \quad \text{trace}\{\mathbf{M}_0\mathbf{W}\} \tag{4a}$$
$$\text{s.t.} \quad \text{trace}\{\mathbf{M}_k\mathbf{W}\} \leq y_k \quad \text{for} \quad k = 1, 2, \ldots, r \tag{4b}$$
$$\mathbf{W} \succeq 0. \tag{4c}$$

One primary objective of this paper is to show that the non-unique conversion from (1) to (2) can be carried out in such a way that the SDP relaxation (4) will have a solution $\mathbf{W}^{\text{opt}}$ with rank at most 2. Also, this solution can be obtained in polynomial time. In other words, the original polynomial optimization (1) is equivalent to the rank-constrained optimization (3) with the property that the latter optimization becomes tractable after relaxing its hard constraint $\text{rank}\{\mathbf{W}\} = 1$ to $\text{rank}\{\mathbf{W}\} \leq 2$. This result has two implications:

i) The NP-hardness of various subclasses of polynomial optimization, e.g., combinatorial optimization, is only related to the existence of a not rank-1 but low-rank SDP solution, where the upper bound on the rank is constant and does not depend on the size of the original optimization.
ii) By approximating the low-rank solution of the SDP relaxation with a rank-1 matrix, an approximate solution of the original problem may be obtained whose closeness to the global solution can also be upper bounded.

These results offer a new insight into the computational complexity of polynomial optimization (Property (i)) and enable to seek a near-global solution (Property (ii)).

A. Related Work

The SDP relaxation technique provides a lower bound on the minimum cost of the original problem, which can be used for various proposes such as the branch and bound algorithm [2]. To understand the quality of the SDP relaxation, its optimal objective value is shown to be at most 14% different from the optimal cost for the MAXCUT problem [26].

The maximum possible gap between the solution of a graph optimization and its SDP relaxation is defined as the Grothendieck constant of the graph [27], [28]. This constant has been derived for some special cases in [29]. The paper [30] shows how a complex SDP relaxation may solve the max-cut problem. This approach has been generalized in several papers [11]–[18]. If the SDP relaxation provides the same optimal objective value as the original problem, the relaxation is said to be exact. The exactness of the SDP relaxation has been verified for a variety of problems [19], [31]–[34]. For instance, our work [19], [35]–[37] has explored the SDP relaxation for the optimal power flow (OPF) problem, which is regarded as the most fundamental optimization problem for electrical power networks. Our work shows that the relaxation is exact for a large class of OPF problems due to the physics of a power gird. The exactness of an SDP relaxation could be heavily formulation dependent. Indeed, we have designed a practical circuit optimization with four equivalent QCQPs in [20], where only one of the formulations has an exact SDP relaxation.

In the case where the SDP relaxation is not exact, the existence of a low-rank SDP solution may still be helpful. To support this claim, we have proposed a penalized SDP relaxation for the OPF problem in [20] and successfully used it to derive near global solutions for 7000 instances of OPF. In a general context, the existence of a low-rank solution to matrix optimizations with linear and LMI constraints has been extensively studied in the literature [21], [22]. The papers [23], [38], [39] provide an upper bound on the lowest rank among all solutions of a feasible LMI problem. Based on the same approach, a constructive method has been proposed in [40] to obtain a low-rank solution in polynomial time. Although the proven bound in [39] is tight in the worst case, many examples are known to possess solutions with a lower rank due to their underlying sparsity patterns [41], [42]. A rank-1 matrix decomposition technique is developed in [24] to find a rank-1 solution whenever the number of constraints is small. This technique is extended in [25] to the complex SDP problem. The paper [40] presents a polynomial-time algorithm for finding an approximate low-rank solution.

This paper is in part related to our recent work [33], [34] that studies the exactness of the SDP relaxation through a graph-theoretic approach. In that work, the structure of an arbitrary polynomial optimization is mapped into a generalized weighted graph, where the topology of the graph captures the sparsity of the optimization and each edge is associated with a weight set capturing possible patterns in the coefficients of the optimization. It is shown that the SDP relaxation is exact if its underlying generalized weighted graph satisfies some algebraic properties.

Given an arbitrary polynomial optimization problem, our technical approach consists of the following three steps:

- **Quadratic Formulation:** First, the polynomial optimization (1) is converted to an equivalent QCQP problem of the form (2). Then, the zero/nonzero pattern of the matrices $\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_r$ is mapped into a simple graph $\mathcal{G}$. The notion of treewidth is adopted to quantify the sparsity level of this non-unique quadratic formulation.
- **Low-rank Solution Recovery:** Since the SDP relaxation of the obtained QCQP does not have a unique solution in general, consider an arbitrary SDP solution. Given this SDP solution and an arbitrary tree decomposition of $\mathcal{G}$ with width t, a convex optimization is designed such that every solution of this problem is a solution of the SDP relaxation and has rank at most $t + 1$.
- **Graph Sparsification:** Finally, three different methods are proposed to sparsify the QCQP problem by increasing its dimension and yet maintaining the equivalence. This will lead to a sparse QCQP formulation for the polynomial optimization (1) whose associated graph has a small treewidth. The sparsification process guarantees that the SDP relaxation of the resulting sparse QCQP has a rank 1 or 2 matrix solution. This low-rank solution can be obtained in polynomial time via the aforementioned low-rank solution recovery technique.

B. Notations

The notations used throughout this paper are described here. $\mathbb{R}$, $\mathbb{Z}_+$, $\mathbb{S}^n$ and $\mathbb{H}^n$ denote the sets of real numbers, nonnegative integer numbers, $n \times n$ symmetric matrices, and $n \times n$ Hermitian matrices, respectively. $\mathbb{S}^n_+$ and $\mathbb{H}^n_+$ denote the restrictions of $\mathbb{S}^n$ and $\mathbb{H}^n$ to positive semidefinite matrices. $\mathrm{Re}\{\mathbf{W}\}$, $\mathrm{Im}\{\mathbf{W}\}$, $\mathrm{rank}\{\mathbf{W}\}$, and $\mathrm{trace}\{\mathbf{W}\}$ denote the real part, imaginary part, rank, and trace of a given scalar/matrix $\mathbf{W}$, respectively. The notation $\mathbf{W} \succeq 0$ means that $\mathbf{W}$ is Hermitian and positive semidefinite. Given a matrix $\mathbf{W}$, its (l, m)-th entry is denoted as W_{lm}. Likewise, the i-th enetry of a vector $\mathbf{x}$ is denoted as x_i. The notation $\mathbf{e}_i$ denotes the i-th ordinary Cartesian unit vector in $\mathbb{R}^n$. $\mathbb{R}[\mathbf{x}]$ denotes the set of all polynomials with the variable $\mathbf{x}$ and real coefficients. The superscript $(.)^{\text{opt}}$ is used to show the globally optimal value of an optimization parameter. The symbol $(\cdot)^*$ represents the conjugate transpose operator. The notation $|x|$ denotes the size of a vector x.

The set of vertices and edges on a directed/underected graph $\mathcal{G}$ is shown by $\mathcal{V}_\mathcal{G}$ and $\mathcal{E}_\mathcal{G}$ respectively. For two simple graphs $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$, the notation $\mathcal{G}_1 \subseteq \mathcal{G}_2$ means that $\mathcal{V}_1 \subseteq \mathcal{V}_2$ and $\mathcal{E}_1 \subseteq \mathcal{E}_2$. $\mathcal{G}_1$ is called a subgraph of $\mathcal{G}_2$ and $\mathcal{G}_2$ is called a supergraph of $\mathcal{G}_1$. A subgraph $\mathcal{G}_1$ of $\mathcal{G}_2$ is said to be an induced subgraph if for every pair of vertices $v_l, v_m \in \mathcal{V}_1$, the relation $(v_l, v_m) \in \mathcal{E}_1$ holds if and only if $(v_l, v_m) \in \mathcal{E}_2$. In this case, $\mathcal{G}_1$ is said to be induced by the vertex subset $\mathcal{V}_1$. The notation $\mathcal{G}_1 \cup \mathcal{G}_2$ also refers to the graph $\mathcal{G} = (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{E}_1 \cup \mathcal{E}_2)$.

II. Quadratic Formulation

The objective of this section is twofold. First, a systematic method will be proposed to formulate the polynomial optimization (1) as a QCQP. Second, a rank-constrained formulation of the problem will be studied using the bisection method and the SDP relaxation technique.

Every polynomial optimization admits infinitely many quadratic formulations. To delve into this property, a simple illustrative example will be provided below.

Example 1. *Assume that $q = 3$ and $r = 2$. Consider the polynomial optimization* (1) *with the objective function*

$$P_0(\mathbf{x}) \triangleq x_1^3 + x_2^2 + 3x_1x_2x_3 \tag{5a}$$

and the constraints

$$P_1(\mathbf{x}) \triangleq x_1^2 - 1 \leq 0 \tag{5b}$$
$$P_2(\mathbf{x}) \triangleq x_3^2 - 1 \leq 0. \tag{5c}$$

Define $U(\mathbf{x})$ as the vector of monomials

$$U(\mathbf{x}) \triangleq [1 \quad x_1 \quad x_2 \quad x_3 \quad x_1^2 \quad x_1x_2]^T. \tag{6}$$

The polynomials $P_0(\mathbf{x})$, $P_1(\mathbf{x})$ and $P_2(\mathbf{x})$ can be expressed as quadratic functions of the entries of $U(\mathbf{x})$:

$$P_0(\mathbf{x}) = U_2(\mathbf{x})U_5(\mathbf{x}) + U_3(\mathbf{x})^2 + 3\ U_4(\mathbf{x})U_6(\mathbf{x}) \tag{7a}$$
$$P_1(\mathbf{x}) = U_2(\mathbf{x})^2 - U_1(\mathbf{x})^2 \tag{7b}$$
$$P_2(\mathbf{x}) = U_4(\mathbf{x})^2 - U_1(\mathbf{x})^2 \tag{7c}$$

On the other hand, the function $U(\mathbf{x})$ is invertible, meaning that $\mathbf{x}$ can be uniquely recovered from $U(\mathbf{x})$. Moreover, it can be shown that the image of the function $U(\mathbf{x})$ is equal to the set of all vectors $\mathbf{u} \in \mathbb{R}^6$ satisfying the following conditions:

$$u_1 = 1 \tag{8a}$$
$$u_5u_1 - u_2^2 = 0 \tag{8b}$$
$$u_6u_1 - u_2u_3 = 0 \tag{8c}$$

It can be concluded from the abovementioned facts that the polynomial optimization (1) *is equivalent to the QCQP problem*

$$\min_{\mathbf{u} \in \mathbb{R}^6} \quad u_2u_5 + u_3^2 + 3u_4u_6 \tag{9a}$$
$$\text{s.t.} \quad u_2^2 - u_1^2 \leq 0 \tag{9b}$$
$$u_4^2 - u_1^2 \leq 0 \tag{9c}$$
$$u_5u_1 - u_2^2 = 0 \tag{9d}$$
$$u_6u_1 - u_2u_3 = 0 \tag{9e}$$
$$u_1 = 1 \tag{9f}$$

In particular, every feasible point $\mathbf{x} \in \mathbb{R}^3$ of (1) *can be mapped into a feasible point $\mathbf{u} \in \mathbb{R}^6$ of* (9) *and vice versa. The above QCQP is referred to as a* **quadratic formulation** *of the polynomial optimization* (1) .

To generalize the idea manifested in Example 1, consider a set of polynomials

$$Q_1(\mathbf{x}), Q_2(\mathbf{x}), \ldots, Q_p(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$$

and a set of matrices

$$\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_p, \mathbf{P}_0, \mathbf{P}_1, \ldots, \mathbf{P}_l \in \mathbb{S}^n$$

such that

$$P_k(\mathbf{x}) = U(\mathbf{x})^T \mathbf{P}_k U(\mathbf{x}), \qquad k = 0, 1, \ldots, l \tag{10a}$$
$$Q_i(\mathbf{x}) = U(\mathbf{x})^T \mathbf{Q}_i U(\mathbf{x}) \qquad i = 1, 2, \ldots, p \tag{10b}$$

where

$$U(\mathbf{x}) \triangleq [1 \quad \mathbf{x}^T \quad Q_1(\mathbf{x}) \quad Q_2(\mathbf{x}) \quad \cdots \quad Q_p(\mathbf{x})]^T \tag{11}$$

and $n = p + q + 1$ denotes the size of the parametric vector $U(\mathbf{x})$. Equation (10a) is indeed a quadratic representation of the polynomial $P_k(\mathbf{x})$ in terms of the entries of $U(\mathbf{x})$. Moreover, equation (10b) describes the interrelation between the entries of $U(\mathbf{x})$ (e.g., see (8)).

Definition 1. *For a triplet $\left(\{Q_i(\mathbf{x})\}_{i=1}^p, \{\mathbf{Q}_i\}_{i=1}^p, \{\mathbf{P}_k\}_{k=0}^l\right)$ satisfying the relations given in* (10)*, define a simple directed graph $\mathcal{H}$ denoted as the* **dependency graph***, whose set of vertices is*

$$\mathcal{V}_\mathcal{H} \triangleq \{U_1(\mathbf{x}), U_2(\mathbf{x}), \ldots, U_n(\mathbf{x})\} \tag{12}$$

and for every $1 \leq i, j \leq n$, the directed edge $(U_i(\mathbf{x}), U_j(\mathbf{x}))$

belongs to $\mathcal{E}_{\mathcal{H}}$ if and only if

$$j > q+1 \quad \text{and} \quad \|\mathbf{Q}_{j-q-1}\mathbf{e}_i\|_2 \neq 0. \tag{13}$$

Then $\left(\{Q_i(\mathbf{x})\}_{i=1}^p, \{\mathbf{Q}_i\}_{i=1}^p, \{\mathbf{P}_k\}_{k=0}^l\right)$ *is called a* **quadratic representation** *for* $\{P_k(\mathbf{x})\}_{k=0}^l$ *if*

a) *$\mathcal{H}$ is an acyclic directed graph with no loops.*
b) *The set of root vertices of $\mathcal{H}$ is equal to $\{U_1(\mathbf{x}), ..., U_{q+1}(\mathbf{x})\}$ or equivalently $\{1, x_1, x_2, \dots, x_q\}$.*

The number n is called the dimension of the quadratic representation.

Lemma 1. *Suppose that* $\left(\{Q_i(\mathbf{x})\}_{i=1}^p, \{\mathbf{Q}_i\}_{i=1}^p, \{\mathbf{P}_k\}_{k=0}^l\right)$ *is a quadratic representation for* $\{P_k(\mathbf{x})\}_{k=0}^l$. *The polynomial optimization* (1) *is equivalent to the QCQP problem:*

$$\min_{\mathbf{u}\in\mathbb{R}^n} \quad \mathbf{u}^T\mathbf{P}_0\mathbf{u} \tag{14a}$$
$$\text{s.t.} \quad \mathbf{u}^T\mathbf{P}_k\mathbf{u} \leq 0 \qquad \text{for} \quad k = 1,2,\dots,l \tag{14b}$$
$$\mathbf{u}^T\mathbf{Q}_i\mathbf{u} = \mathbf{e}_{i+q+1}^T\mathbf{u} \quad \text{for} \quad i = 1,2,\dots,p \tag{14c}$$
$$u_1 = 1, \tag{14d}$$

In addition, every globally optimal solution $\mathbf{x}^{opt}$ can be mapped to an optimal solution $\mathbf{u}^{opt}$, and vice versa, through the equation $\mathbf{u}^{opt} = U(\mathbf{x}^{opt})$.

Proof. It follows from Property (b) in Definition 1 and equation (10b) that for every $j > q+1$, $U_j(\mathbf{x})$ (or equivalently $Q_{j-q-1}(\mathbf{x})$) can be uniquely expressed in terms of the members of the set

$$\{U_i(\mathbf{x}) \,|\, U_i(\mathbf{x}) \rightarrow U_j(\mathbf{x}),\, 1 \leq i \leq n\}. \tag{15}$$

With no loss of generality, assume that $U_i(\mathbf{x})$ is a root vertex induced by $\{U_i(\mathbf{x}), U_{i+1}(\mathbf{x}), \dots, U_n(\mathbf{x})\}$ for every $i \in \{2+q, ..., n\}$.

By induction and using equation (10b), it can be shown that $U_i(\mathbf{x})$ can be expressed as a function of the elements of $\{U_1(\mathbf{x}), U_2(\mathbf{x}), \dots, U_{i-1}(\mathbf{x})\}$ for $i \in \{2+q, ..., n\}$. Similarly, u_i can be expressed as the same function of the elements of $\{u_1, u_2, \dots, u_{i-1}\}$, provided $\mathbf{u}$ is a feasible solution of Optimization (14). This implies that for every feasible solution $\mathbf{u}$, there is a vector $\mathbf{x}$ such that $\mathbf{u} = U(\mathbf{x})$. This completes the proof. □

Lemma 2. *Suppose that $\{P_k(\mathbf{x})\}_{k=0}^l$ are polynomials of degree at most d, consisting of s monomials in total. There exists a quadratic representation of $\{P_k(\mathbf{x})\}_{k=0}^l$ with dimension n, where*

$$n \leq 1 + q \times s \times (\lfloor \log_2(d) \rfloor + 2) \tag{16}$$

Proof. For every natural number k, the monomial x_i^k can be written as the product of some distinct members of the set

$$A_i \triangleq \{x_i^{2^j} \,|\, j \in \{0, \dots, \lfloor \log_2(d) \rfloor + 1\}\} \tag{17}$$

This can be achieved using the binary expansion of the exponent k. Let A be a set of monomials defined as $A_1 \cup \dots \cup A_n$. Consider a monomial $x_1^{\alpha_1}x_2^{\alpha_2}\dots x_q^{\alpha_q}$ with degree at most d (i.e., $\alpha_1 + \dots + \alpha_q \leq d$). This monomial can be written as the product $a_1a_2\dots a_c$, where $a_1, a_2, \dots, a_c$ are some distinct monomial members of A and $c \leq q\left(\lfloor \log_2(d) \rfloor + 1\right)$. Define

$$b_1 \triangleq a_1a_2, \;\; b_2 \triangleq a_1a_2a_3, \;\; \dots \;\; , \;\; b_{c-1} \triangleq a_1a_2\dots a_c \tag{18}$$

Let $U(\mathbf{x})$ be a vector containing number 1, all members of the set A, and the vector $[b_1 \;\cdots\; b_{c-1}]$ for each of the s monomials $x_1^{\alpha_1}x_2^{\alpha_2}\dots x_q^{\alpha_q}$'s in $\{P_k(\mathbf{x})\}_{k=0}^l$. Notice that the size of the vector $U(\mathbf{x})$, denoted as n, satisfies the inequality (16). On the other hand, one can write

$$b_{c-1} = x_1^{\alpha_1}x_2^{\alpha_2}\dots x_q^{\alpha_q} \tag{19a}$$
$$b_i = b_{i-1}a_{i+1}, \qquad\qquad i = 2, ..., c-1 \tag{19b}$$

Using the above recursions, it is to verify the existence of a quadratic representation associated with the vector $U(\mathbf{x})$. □

By combining Lemmas 1 and 2, it can be inferred that the polynomial optimization (1) can be equivalently converted to a QCQP problem of the form (14) after introducing a modest number of auxiliary variables and constraints. This QCQP problem can also be transformed into the form (2) containing no linear terms in the objective and constraints. To achieve this goal, two operations are required:

- Replace the equality constraint (14c) with two inequality constraints:

$$\mathbf{u}^T(2\mathbf{Q}_i - \mathbf{e}_1\mathbf{e}_{i+q+1}^T - \mathbf{e}_{i+q+1}\mathbf{e}_1^T)\mathbf{u} \leq 0 \tag{20a}$$
$$\mathbf{u}^T(-2\mathbf{Q}_i + \mathbf{e}_1\mathbf{e}_{i+q+1}^T + \mathbf{e}_{i+q+1}\mathbf{e}_1^T)\mathbf{u} \leq 0 \tag{20b}$$

- Replace the linear constraint (14d) with

$$\mathbf{u}^T(\mathbf{e}_1\mathbf{e}_1^T)\mathbf{u} \leq 1 \quad \text{and} \quad \mathbf{u}^T(-\mathbf{e}_1\mathbf{e}_1^T)\mathbf{u} \leq -1 \tag{21}$$

  Although these two constraints are equivalent to $u_1^2 = 1$ as opposed to $u_1 = 1$, since

$$\mathbf{u}^T\mathbf{M}_k\mathbf{u} = (-\mathbf{u})^T\mathbf{M}_k(-\mathbf{u}), \tag{22}$$

  the sign of u_1 is not important.

A. Rank-Constrained Formulation

So far, it has been shown in the preceding subsection that an arbitrary polynomial Optimization (1) can be formulated as a QCQP problem (2) by introducing a modest number of auxiliary variables and additional constraints (the problem description still has a polynomial size). We may also proceed with a linear representation of the constraints and objective function in (2) by defining a new matrix variable $\mathbf{W} \in \mathbb{S}^n$. To this end, define

$$\mathbf{W} \triangleq \mathbf{u}\mathbf{u}^T. \tag{23}$$

Every quadratic term $\mathbf{u}^T\mathbf{M}_k\mathbf{u}$ has a linear representation with respect to $\mathbf{W}$ as follows:

$$\begin{aligned}\mathbf{u}^T\mathbf{M}_k\mathbf{u} &= \text{trace}\{\mathbf{u}^T\mathbf{M}_k\mathbf{u}\} \\ &= \text{trace}\{\mathbf{M}_k\mathbf{u}\mathbf{u}^T\} \\ &= \text{trace}\{\mathbf{M}_k\mathbf{W}\}, \quad k = 0,1,...,r.\end{aligned} \tag{24}$$

On the other hand, an arbitrary matrix $\mathbf{W} \in \mathbb{S}^n$ can be factorized as $\mathbf{u}\mathbf{u}^T$ for some vector $\mathbf{u}$ if $\mathbf{W} \succeq 0$ and $\text{rank}\{\mathbf{W}\} = 1$. Therefore, it can be concluded that the optimization problem

(1) and its quadratic formulation (2) admit a rank-constrained formulation:

$$\min_{\mathbf{W}\in\mathbb{S}^n} \quad \text{trace}\{\mathbf{M}_0\mathbf{W}\} \tag{25a}$$
$$\text{s.t.} \quad \text{trace}\{\mathbf{M}_k\mathbf{W}\} \leq y_k \quad \text{for} \quad k=1,2,\dots,r \tag{25b}$$
$$\mathbf{W} \succeq 0 \tag{25c}$$
$$\text{rank}\{\mathbf{W}\} = 1 \tag{25d}$$

B. Feasibility Problem and Bisection Approach

In order to solve the rank-constrained optimization (25), we exploit the well-known epigraph technique to translate the objective of this optimization into a suitable constraint. For this purpose, consider the problem

$$\min_{y_0\in\mathbb{R},\, \mathbf{W}\in\mathbb{S}^n} \quad y_0 \tag{26a}$$
$$\text{s.t.} \quad \text{trace}\{\mathbf{M}_k\mathbf{W}\} \leq y_k \quad \text{for} \quad k=0,1,\dots,r \tag{26b}$$
$$\mathbf{W} \succeq 0 \tag{26c}$$
$$\text{rank}\{\mathbf{W}\} = 1 \tag{26d}$$

(note that y_0 is a variable, whereas $y_1, ..., y_r$ are all known). It is trivial that the constraint

$$\text{trace}\{\mathbf{M}_0\mathbf{W}\} \leq y_0 \tag{27}$$

is active for every optimal solution $(y_0^{\text{opt}}, \mathbf{W}^{\text{opt}})$, meaning that

$$\text{trace}\{\mathbf{M}_0\mathbf{W}^{\text{opt}}\} = y_0^{\text{opt}}. \tag{28}$$

Therefore, y_0^{opt} can be interpreted as the optimal cost for the rank-constrained optimization (25). To solve this problem using the reformulation (26), define

$$\mathcal{S}\left(\mathbf{M}_0, \mathbf{M}_1, \dots, \mathbf{M}_r;\, y_0, y_1, \dots, y_r\right) \triangleq$$
$$\Big\{\mathbf{W}\in\mathbb{S}_+^n \;\Big|\; \text{trace}\{\mathbf{M}_k\mathbf{W}\} \leq y_k \;\; \text{for} \;\; k=0,1,\dots,r\Big\} \tag{29}$$

Every matrix $\mathbf{W}$ in the above set provides an upper bound

$$y_0^+ \triangleq \text{trace}\{\mathbf{M}_0\mathbf{W}\}. \tag{30}$$

on y_0^{opt} (i.e., the solution of the polynomial optimization (1)). A lower bound y_0^- may also be provided using the SDP relaxation method. Let $\epsilon_0 \triangleq y_0^+ - y_0^-$ and ϵ be the tolerance of error for finding the optimal objective value. It is straightforward to argue that the bisection method provided below results in a feasible point $(\hat{y}_0, \widehat{\mathbf{W}})$ for problem (25) with a cost ϵ-close to the global minimum.

Bisection Method:

1) Set
$$i := 0, \quad y^+ := y_0^+ \quad \text{and} \quad y^- := y_0^- \tag{31}$$

2) If there exists a rank-1 matrix $\widehat{\mathbf{W}}$ in the set
$$\mathcal{S}\left(\mathbf{M}_0, \mathbf{M}_1, \dots, \mathbf{M}_r;\, \frac{y^+ + y^-}{2}, y_1, \dots, y_r\right),$$
then update the upper bound as
$$y^+ := \frac{y^+ + y^-}{2}, \tag{32}$$
otherwise, update the lower bound as
$$y^- := \frac{y^+ + y^-}{2}. \tag{33}$$

3) If
$$i \leq \lceil \log_2(\epsilon_0/\epsilon) \rceil, \tag{34}$$
then set $i := i+1$ and go to Step 2. Otherwise, set
$$\hat{y} := y^+, \tag{35}$$
and terminate.

The number of iterations needed in Bisection Method is logarithmic, where the existence of a rank-1 matrix in the convex set $\mathcal{S}$ should be verified in each iteration. Hence, this method enlightens the equivalence of solving an arbitrary polynomial optimization (1) and the problem of searching for a rank-1 member of $\mathcal{S}$. In other words, the complexity of solving a polynomial optimization can be traced back to checking the existence of a rank-1 matrix a modest number of times (namely, $\lceil \log_2(\epsilon_0/\epsilon) \rceil$). Since this membership verification is NP-hard in the worst case, the following question arises: what is the smallest number k for which the existence of a matrix $\mathbf{W} \in \mathcal{S}$ with the property $\text{rank}\{\mathbf{W}\} \leq k$ could be checked in polynomial time? To address this question, one objective of this paper is to show that the polynomial optimization (1) has a quadratic formulation associated with some sparse matrices $M_0, ..., M_r$ such that its corresponding set $\mathcal{S}$ can be searched for a rank-2 matrix in polynomial time. In other words, the smallest number k is equal to 2.

C. SDP Relaxation

To reduce the complexity of the rank-constrained optimization (25), one may drop its rank constraint $\text{rank}\{\mathbf{W}\} = 1$. The resulting convex program is called an SDP relaxation of the QCQP problem (2). The relaxation is said to be exact if the QCQP problem and its relaxation have the same optimal value. Under this circumstance, the relaxation has a rank-1 solution. In the case where the relaxation is not exact, it is highly desirable to explore the rank of the minimum-rank solution of the SDP relaxation. As long as this is a small number, a penalization method may be able to produce a near-optimal rank-1 SDP solution (see [43] and [20]). Thus, the following question arises: what is the smallest number k for which the SDP relaxation of the QCQP problem (2) has a solution $\mathbf{W}^{\text{opt}}$ with the property $\text{rank}\{\mathbf{W}^{\text{opt}}\} \leq k$? It will be shown later in this work that $k = 2$. In other words, there exists a quadratic formulation whose SDP relaxation has a rank 1 or 2 solution. In addition, this possibly hidden low-rank solution can be found in polynomial time.

III. Low-Rank Solution Recovery

The SDP relaxation of a sparse QCQP problem may have many matrix solutions including low-rank and high-rank solutions. The objective of this section is twofold. First, the connection between the rank of a low-rank SDP solution and the sparsity level of the problem is studied. Second, a convex program is proposed to find a low-rank solution (if such a

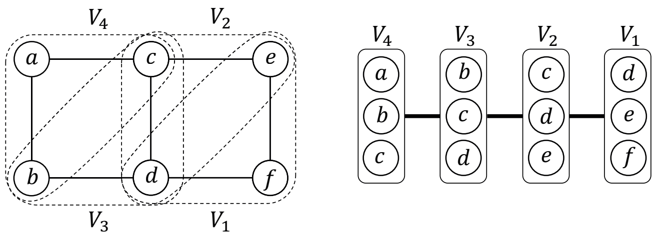

Fig. 1: A minimal tree decomposition for a ladder

solution exists). To proceed with this part, we first review some preliminaries about the tree decomposition of a graph.

A. Preliminaries on Graph Theory

Definition 2. *The representative graph of an $n \times n$ symmetric matrix $\mathbf{W}$, denoted by $\mathcal{G}(\mathbf{W})$, is a simple graph with n vertices whose edges are specified by the locations of the nonzero off-diagonal entries of $\mathbf{W}$. In other words, two arbitrary vertices i and j are connected if W_{ij} is nonzero.*

Definition 3 (Treewidth)**.** *Given a graph $\mathcal{G}$, a tree $\mathcal{T}$ is called a tree decomposition of $\mathcal{G}$ if it satisfies the following properties:*

1) *Every node of $\mathcal{T}$ corresponds to and is identified by a subset of $\mathcal{V}_{\mathcal{G}}$. Alternatively, each node of $\mathcal{T}$ is regarded as a group of vertices of $\mathcal{G}$.*
2) *Every vertex of $\mathcal{G}$ is a member of at least one node of $\mathcal{T}$.*
3) *$\mathcal{T}_k$ is a connected graph for $k = 1, 2, ..., n$, where $\mathcal{T}_k$ denotes the subgraph of $\mathcal{T}$ induced by all nodes of $\mathcal{T}$ containing the vertex k of $\mathcal{G}$.*
4) *The subgraphs $\mathcal{T}_i$ and $\mathcal{T}_j$ have a node in common for every $(i, j) \in \mathcal{E}_{\mathcal{G}}$.*

The width of a tree decomposition is the cardinality of its biggest node minus one (recall that each node of $\mathcal{T}$ is indeed a set containing a number of vertices of $\mathcal{G}$). The treewidth of $\mathcal{G}$ is the minimum width over all possible tree decompositions of $\mathcal{G}$ and is denoted by $\mathrm{tw}(\mathcal{G})$.

Note that the treewidth of a tree is equal to 1. Figure 1 shows a graph $\mathcal{G}$ with 6 vertices named a, b, c, d, e, f, together with its minimal tree decomposition $\mathcal{T}$. Every node of $\mathcal{T}$ is a set containing three members of $\mathcal{V}_{\mathcal{G}}$. The width of this decomposition is therefore equal to 2.

Definition 4 (Enriched Supergraph)**.** *Given a graph $\mathcal{G}$ accompanied by a tree decomposition $\mathcal{T}$ of width t, $\overline{\mathcal{G}}$ is called an enriched supergraph of $\mathcal{G}$ derived by $\mathcal{T}$ if it is obtained according to the following procedure:*

1) *Add a sufficient number of (redundant) vertices to the nodes of $\mathcal{T}$, if necessary, in such a way that every node includes exactly $t+1$ vertices. Also, add the same vertices to $\mathcal{G}$ (without incorporating new edges). Denote the new graphs associated with $\mathcal{T}$ and $\mathcal{G}$ as $\tilde{\mathcal{T}}$ and $\tilde{\mathcal{G}}$, respectively.*
2) *Index the nodes of the tree $\tilde{\mathcal{T}}$ as $V_1, V_2, \ldots, V_{|\mathcal{T}|}$ in such a way that for every $r \in \{1, ..., |\mathcal{T}|\}$, the node V_r becomes a leaf of $\mathcal{T}^r$ defined as the subgraph of $\tilde{\mathcal{T}}$ induced by $\{V_1, \ldots, V_r\}$. Denote the neighbor of V_r in $\mathcal{T}^r$ as $V_{r'}$ (note that $V_r \in \mathcal{V}_{\mathcal{G}}$).*
3) *Set $\mathcal{G}^{|\mathcal{T}|} := \tilde{\mathcal{G}}$ and set $\mathcal{O}^{|\mathcal{T}|}$ as the empty sequence. Also set $k = |\mathcal{T}|$.*
4) *Let $V_k \backslash V_{k'} = \{o_1, \ldots, o_s\}$ and $V_{k'} \backslash V_k = \{w_1, \ldots, w_s\}$. Set*

$$\mathcal{G}^{k-1} := (\mathcal{V}_{\mathcal{G}^k}, \mathcal{E}_{\mathcal{G}^k} \cup \{(o_1, w_1), \ldots, (o_s, w_s)\}) \quad (36)$$
$$\mathcal{O}^{k-1} := \mathcal{O}^k \cup (o_1, \ldots, o_s) \quad (37)$$
$$k := k - 1 \quad (38)$$

5) *If $k = 1$ set $\overline{\mathcal{G}} := \mathcal{G}^1$, $\mathcal{O} := \mathcal{O}^1$ and terminate; otherwise go to step 4. $\overline{\mathcal{G}}$ is referred to as an enriched supergraph of $\mathcal{G}$ derived by $\mathcal{T}$.*

Figure 2 delineates the process of obtaining an enriched supergraph $\overline{\mathcal{G}}$ of the graph $\mathcal{G}$ depicted in Figure 1. Bold lines show the edges added at each step of the algorithm.

B. Low-rank Recovery via Convex Optimization

Consider the QCQP problem (2). Let

$$\mathcal{G} \triangleq \mathcal{G}(M_0) \cup \mathcal{G}(M_1) \cup \ldots \cup \mathcal{G}(M_r) \quad (39)$$

be defined as the sparsity graph associated with the problem (2). The graph $\mathcal{G} = (\mathcal{V}_{\mathcal{G}}, \mathcal{E}_{\mathcal{G}})$ describes the zero-nonzero pattern of the matrices $\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_r$. This graph can also be interpreted as follows:

- There is a one to one mapping between the vertices of $\mathcal{G}$ and the entries of the variable $\mathbf{u}$ in (2). This means that for every $1 \leq i \leq n$, there is a unique vertex $v \in \mathcal{V}_{\mathcal{G}}$ corresponding to u_i. We adopt the notation $v \leftrightarrow u_i$ to show the correspondence.
- For every pair of distinct vertices $v_1, v_2 \in \mathcal{V}_{\mathcal{G}}$, we have $(v_1, v_2) \in \mathcal{E}_{\mathcal{G}}$ if and only the product $u_i u_j$ has a nonzero coefficient in at least one of the constraints (2b) or the objective function (2a) where $v_1 \leftrightarrow u_i$ and $v_2 \leftrightarrow u_j$.

Let $\bar{\mathcal{G}}$ be an enriched supergraph of $\mathcal{G}$, obtained from a tree decomposition of width t. For simplicity, we label the vertices of $\mathcal{G}$ and $\bar{\mathcal{G}}$ as

$$\mathcal{V}_{\mathcal{G}} = \{1, 2, \ldots, n\} \quad \text{and} \quad \mathcal{V}_{\bar{\mathcal{G}}} = \{1, 2, \ldots, m\}. \quad (40)$$

such that $\mathcal{E}(\mathcal{G}) \subseteq \mathcal{E}(\bar{\mathcal{G}})$.

Theorem 1. *Consider an arbitrary solution $\widehat{\mathbf{W}} \in \mathbb{S}_+^n$ to the SDP relaxation problem* (4) *and let $\mathbf{Z} \in \mathbb{S}^m$ be a matrix with the property that $\mathcal{G}(\mathbf{Z}) = \bar{\mathcal{G}}$. Let $\overline{\mathbf{W}}^{\text{opt}}$ denote an arbitrary solution of the optimization*

$$\min_{\overline{\mathbf{W}} \in \mathbb{S}^m} \quad \text{trace}\{\mathbf{Z}\overline{\mathbf{W}}\} \quad (41a)$$
$$\text{s.t.} \quad \overline{W}_{kk} = \widehat{W}_{kk} \quad \text{for} \quad k \in \mathcal{V}_{\mathcal{G}}, \quad (41b)$$
$$\overline{W}_{kk} = 1 \quad \text{for} \quad k \in \mathcal{V}_{\bar{\mathcal{G}}} \setminus \mathcal{V}_{\mathcal{G}}, \quad (41c)$$
$$\overline{W}_{ij} = \widehat{W}_{ij} \quad \text{for} \quad (i, j) \in \mathcal{E}_{\mathcal{G}}, \quad (41d)$$
$$\overline{\mathbf{W}} \succeq 0. \quad (41e)$$

Define $\mathbf{W}^{\text{opt}}$ as the n-th principal minor of $\overline{\mathbf{W}}^{\text{opt}}$. Then, $\mathbf{W}^{\text{opt}}$ satisfies the following two properties:

a) *$\mathbf{W}^{\text{opt}}$ is an optimal solution to the SDP relaxation* (4).
b) $\text{rank}\{\mathbf{W}^{\text{opt}}\} \leq t + 1$.

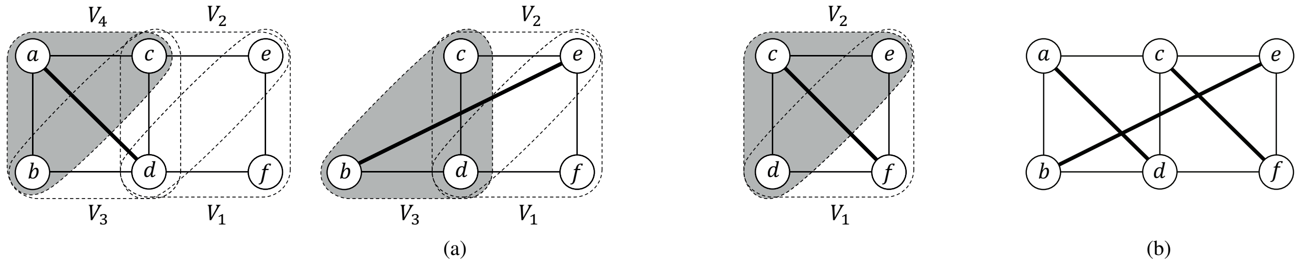

Fig. 2: An enriched supergraph $\overline{\mathcal{G}}$ of the graph $\mathcal{G}$ given in Figure 1: (a) the steps of the algorithm, (b) the resulting enriched supergraph (new edges are shown in bold).

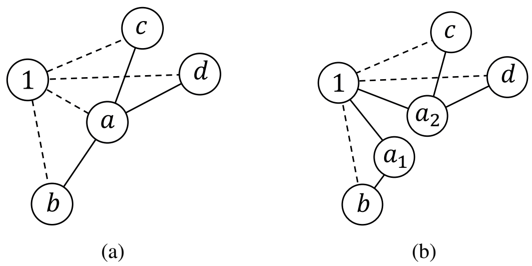

Fig. 3: To sparsify a quadratic formulation, the variable x_a is replaced by two new variables x_{a_1} and x_{a_2}, and then the constraint $x_{a_1} = x_{a_2}$ is added to preserve equivalence: (a) original graph (b) the graph after applying Vertex Duplication Procedure.

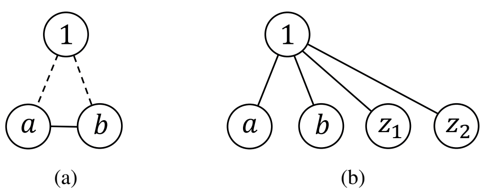

Fig. 4: To sparsify a quadratic formulation, two auxiliary variables x_{z_1} and x_{z_2} are added, and then the product $x_a x_b$ is replaced by $x_{z_1}^2 - x_{z_2}^2$ where $x_{z_1} = (x_a + x_b)/2$ and $x_{z_2} = (x_a - x_b)/2$: (a) original graph (b) the graph after applying Edge Elimination Procedure.

Proof. See [44] for the proof. □

The matrix $\mathbf{W}^{\text{opt}}$ is referred to as a subsolution of (41). Theorem 1 states that a (low-rank) solution with a guaranteed bound on its rank can be constructed from an arbitrary (high-rank) solution of the SDP relaxation (4) by means of the convex optimization (41).

IV. Graph Sparsification

Consider the polynomial optimization (1). It has been shown in Lemmas 1 and 2 that this optimization admits a quadratic formulation. Indeed, it is straightforward to verify that optimization (1) has infinitely many equivalent quadratic formulations. A question arises as to which quadratic formulation is more amenable to the SDP relaxation technique. To address this problem, consider an arbitrary quadratic formulation of the polynomial optimization (1):

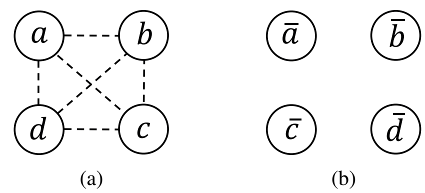

Fig. 5: A change of coordination can remove a clique form the sparsity graph: (a) original graph in which any of the dashed edges may exist, (b) the graph for the reformulated problem in which the vertices corresponding to the new variables $x_{\bar{a}}, x_{\bar{b}}, x_{\bar{c}}, x_{\bar{d}}$ are all isolated.

$$\min_{\tilde{\mathbf{u}} \in \mathbb{R}^{\tilde{n}}} \quad \tilde{\mathbf{u}}^T \tilde{\mathbf{M}}_0 \tilde{\mathbf{u}} \tag{42a}$$

$$\text{s.t.} \quad \tilde{\mathbf{u}}^T \tilde{\mathbf{M}}_k \tilde{\mathbf{u}} \leq \tilde{y}_k \quad \text{for} \quad k = 1, 2, \ldots, \tilde{r} \tag{42b}$$

and let $\tilde{\mathcal{G}} = \mathcal{G}(\tilde{\mathbf{M}}_0) \cup \ldots \cup \mathcal{G}(\tilde{\mathbf{M}}_{\tilde{r}})$ denote the sparsity graph of this optimization. Theorem 1 enables to find a matrix solution for the SDP relaxation of the above QCQP with rank at most $\text{tw}(\tilde{\mathcal{G}}) + 1$. In the case where the matrices $\tilde{\mathbf{M}}_0, \ldots, \tilde{\mathbf{M}}_{\tilde{r}}$ are not sparse, the number $\text{tw}(\tilde{\mathcal{G}})$ is large and hence the obtained SDP solution may not have a good rank-1 approximation. The main objective of this part is to introduce certain operations that can be performed to transform (42) into an equivalent QCQP with a sparse formulation. These operations will be spelled out below, which are categorized as: (i) vertex duplication, (ii) edge elimination, and (iii) change of coordinates.

Notation 1. *Consider a quadratic formulation of optimization* (1) *with the graph $\tilde{\mathcal{G}}$. Given a vertex v in $\tilde{\mathcal{G}}$, the unique variable of the quadratic formulation associated with the vertex v is denoted as x_v.*

A. Vertex Duplication

Our first graph sparsification approach resembles a well-established technique in distributed computation. The main idea is to replace a global variable of the optimization with two local variables and then impose a consistency constraint reflecting the fact that the two new variables are identical copies of the same global variable. This sparsification approach is delineated below.

Vertex Duplication Procedure: Consider an arbitrary quadratic formulation (42). Let a be a vertex of the sparsity graph belonging to a cycle $\mathcal{C}$, where b and c denote the neighboring vertices of a in the cycle $\mathcal{C}$. We perform the following operations:

- Replace the variable x_a with two auxiliary variables x_{a_1} and x_{a_2}.
- Replace the product $x_a x_b$ with $x_{a_1} x_b$ and the product $x_a x_c$ with $x_{a_2} x_c$ in all of the constraints and the objective function of the QCQP.
- For every other neighbor of a, namely the vertex d, replace $x_a x_d$ with either $x_{a_1} x_d$ or $x_{a_2} x_d$ in all of the constraints and the objective function of the QCQP.
- Add the consistency constraint $x_{a_1} = x_{a_2}$ to the quadratic formulation.

Vertex Duplication Procedure is illustrated in Figure 3, where solid lines represent existing edges and dashed lines show possible edges. As can be seen, this procedure manipulates the graph in three ways: (i) replacing the vertex a with two new vertices a_1 and a_2, (ii) distributing the neighbors of a between a_1 and a_2, and (iii) connecting vertex 1 to a_1 and a_2 (vertex 1 corresponds to the first entry of $\tilde{\mathbf{u}}$).

Theorem 2. *Consider an arbitrary quadratic formulation (42) of the polynomial optimization* (1) *associated with a graph $\tilde{\mathcal{G}}$. If Vertex Duplication Procedure is applied to the graph $\tilde{\mathcal{G}}$ a sufficient number of times, it yields a sparse quadratic formulation of the form (2) that is equivalent to optimization* (1) *and its SDP relaxation has a solution $\mathbf{W}^{opt}$ such that*

$$\mathrm{rank}\{\mathbf{W}^{\mathrm{opt}}\} \leq 3 \tag{43}$$

Moreover, this SDP solution can be found in polynomial time.

Proof. Consider a graph with the property that every cycle of the graph passes through a certain vertex v. In this case, all cycles of the graph share one central vertex. It can be verified that such a graph has a trivial tree decomposition of width 2, where each node of the tree consists of two connected vertices of the graph as well as the central vertex v. On the other hand, it can be observed in Figure 3 that Vertex Duplication Procedure makes the vertex 1 a hub if the procedure is repeated a sufficient number of times. This occurs if the underlying sparsification technique is proceeded until the vertex 1 belongs to every cycle of the graph. Let the obtained graph be denoted as $\mathcal{G}$ associated with a sparse QCQP (2). The treewidth of $\mathcal{G}$ is at most 2 and, as a result, Theorem 1 can be deployed to find a solution $\mathbf{W}^{\mathrm{opt}}$ with rank at most 3 for the SDP relaxation of the sparse QCQP (2). □

Theorem 2 derives a sparse QCQP formulation (2) from an arbitrary (dense) quadratic formulation (42). It can be shown that this sparsification increases the number of variables of the optimization by twice the number of edges of $\tilde{\mathcal{G}}$ in the worst case (two new variables for each edge).

B. Edge Elimination

The previous sparsification approach was focused on reducing the degree of each vertex by means of introducing multiple copies of the same vertex. Another approach is to boost the sparsity of the graph by eliminating specific edges of the graph. This idea will be described below.

Edge Elimination Procedure: Consider an arbitrary quadratic formulation (42). Let (a, b) be an edge of its underlying sparsity graph, corresponding to the variables x_a and x_b. We perform the following operations:

- Add two auxiliary variables x_{z_1} and x_{z_2}.
- Impose two additional constraints:

$$x_{z_1} = \frac{x_a + x_b}{2} \quad \text{and} \quad x_{z_2} = \frac{x_a - x_b}{2} \tag{44}$$

- Replace every instance of the product $x_a x_b$ in the QCQP problem with $x_{z_1}^2 - x_{z_2}^2$.

Edge Elimination Procedure is illustrated in Figure 4. As can be seen, this procedure manipulates the graph in three ways: (i) adding two new vertices z_1 and z_2, (ii) removing the edge (a, b), and (iii) connecting vertex 1 to z_1 and z_2.

Theorem 3. *Consider an arbitrary quadratic formulation (42) of the polynomial optimization* (1) *associated with a graph $\tilde{\mathcal{G}}$. If Edge Elimination Procedure is applied to the graph $\tilde{\mathcal{G}}$ a sufficient number of times, it yields a sparse quadratic formulation of the form (2) that is equivalent to optimization* (1) *and its SDP relaxation has a solution $\mathbf{W}^{opt}$ such that*

$$\mathrm{rank}\{\mathbf{W}^{\mathrm{opt}}\} \leq 2 \tag{45}$$

Moreover, this SDP solution can be found in polynomial time.

Proof. The proof is in line with that of Theorem 2. The only difference is that the resulting graph $\mathcal{G}$ will have treewidth 1 as opposed to 2. This is due to the fact that Edge Elimination Procedure produces a star graph with vertex 1 as its central node, and hence this procedure is able to remove all cycles of the graph. □

Corollary 1. *An arbitrary polynomial optimization* (1) *has an equivalent rank-constrained formulation in the form of* (3) *with the property that the relaxation of its constraint* $\mathrm{rank}\{\mathbf{W}\} = 1$ *to* $\mathrm{rank}\{\mathbf{W}\} \leq 2$ *makes the resulting problem polynomial-time solvable.*

Proof. According to Theorem 3, the polynomial optimization (1) has an SDP relaxation with a solution $\mathrm{rank}\{\mathbf{W}^{\mathrm{opt}}\} \leq 2$. This implies that adding the constraint $\mathrm{rank}\{\mathbf{W}\} \leq 2$ to that particular SDP relaxation does not affect the optimal cost. The proof is completed by noting that adding the constraint $\mathrm{rank}\{\mathbf{W}\} = 1$ to the same SDP relaxation makes the problem equivalent to optimization (1). □

C. Change of Coordinates

Consider an arbitrary quadratic formulation (42) of the polynomial optimization (1) associated with a graph $\tilde{\mathcal{G}}$. Assume that at least one of the matrices $\tilde{\mathbf{M}}_0, ..., \tilde{\mathbf{M}}_{\tilde{r}}$ is not sparse and contains a high number of nonzero entries. In this case, the graph $\tilde{\mathcal{G}}$ will have large cliques. Since the treewidth of a graph is lower bounded by the size of its largest clique, the existence of a large clique in the graph $\tilde{\mathcal{G}}$ destroys the low-rank property of the solutions of optimization (41) provided in Theorem 1. Under this circumstance, the vertex duplication and edge elimination procedures described earlier can be utilized to lower the treewidth. Another approach is to change the coordinates of the feasible set of the optimization in such a way that the cliques of the graph all disappear. For example, consider the vertices a, b, c, and d in Figure 5 and assume that some of these vertices are connected to one another. The high-level idea to transform the tuple (x_a, x_b, x_c, x_d) into another set of variables $(x_{\bar{a}}, x_{\bar{b}}, x_{\bar{c}}, x_{\bar{d}})$ such that the vertices become isolated. In what follows, a technique will be proposed to achieve this goal.

Consider the quadratic formulation (42) . Let

$$\tilde{\mathbf{M}}_k = \mathbf{N}_k \mathbf{\Lambda}_k \mathbf{N}_k^T, \quad k = 0, 1, ..., \tilde{r} \tag{46}$$

be the eigendecomposition of $\tilde{\mathbf{M}}_k$, meaning that the i-th column of $\mathbf{N}_k \in \mathbb{R}^{\tilde{n} \times \tilde{n}}$ is a normalized eigenvector of $\tilde{\mathbf{M}}_k$ corresponding to its i-th eigenvalue and $\mathbf{\Lambda}_k \in \mathbb{R}^{\tilde{n} \times \tilde{n}}$ is a diagonal matrix whose i-th diagonal entry is the i-th eigenvalue of $\tilde{\mathbf{M}}_k$, for every $i \in \{1, \ldots, \tilde{n}\}$. Define

$$\tilde{\mathbf{u}}(k) \triangleq \mathbf{N}_k^T \tilde{\mathbf{u}}, \quad k = 0, 1, ..., \tilde{r} \tag{47}$$

and

$$\bar{\mathbf{u}} \triangleq \begin{bmatrix} \tilde{\mathbf{u}}^T & \tilde{\mathbf{u}}(0)^T & \cdots & \tilde{\mathbf{u}}(\tilde{r})^T \end{bmatrix}^T. \tag{48}$$

The next theorem proposes a quadratic formulation with respect to the variable $\bar{\mathbf{u}}$, whose SDP relaxation has a matrix solution of rank 1 or 2.

Theorem 4. *The QCQP problem*

$$\min_{\bar{\mathbf{u}}} \; \tilde{\mathbf{u}}(0)^T \mathbf{\Lambda}_0 \tilde{\mathbf{u}}(0) \tag{49a}$$

$$\text{s.t.} \quad \tilde{\mathbf{u}}(k)^T \mathbf{\Lambda}_k \tilde{\mathbf{u}}(k) \leq \tilde{y}_k \quad \text{for} \quad k = 1, 2, \ldots, \tilde{r} \tag{49b}$$

$$\tilde{\mathbf{u}}(k) = \mathbf{N}_k^T \tilde{\mathbf{u}} \quad \text{for} \quad k = 0, 1, \ldots, \tilde{r} \tag{49c}$$

(subject to the implicit constraint (48)*) is a quadratic formulation of the polynomial optimization* (1) *with the property that its SDP relaxation has a rank 1 or 2 solution that can be obtained in polynomial time.*

Proof. The QCQP problem (49) can be cast as the standard QCQP (2). Let $\mathcal{G}$ denote the sparsity graph of this formulation. Notice that since the matrices $\mathbf{\Lambda}_0, \mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_{\tilde{r}}$ are all diagonal, the graph $\mathcal{G}$ does not have any edge created by the objective function (49a) or the constraint (49b). This implies that all edges of $\mathcal{G}$ are due to the constraint (49c). On the other hand, the constraint (49c) can be expressed in a quadratic form as

$$\tilde{\mathbf{u}}^T \mathbf{R}_k \tilde{\mathbf{u}} = 0, \tag{50}$$

where $\mathbf{R}_k$ is a square matrix of appropriate dimension with the property that every entry of $\mathbf{R}_k$ not belonging to the first row or first column of this matrix is equal to zero. As a result, the constraint (49c) can only yield edges in $\mathcal{G}$ that are incident to the vertex corresponding to 1 (the first entry of $\bar{\mathbf{u}}$). This implies that every edge of the graph $\mathcal{G}$ passes through the vertex 1. Therefore, $\mathcal{G}$ is a tree and $\mathrm{tw}(\mathcal{G}) = 1$. Now, it follows from Theorem 1 that the SDP relaxation of (49) has a rank 1 or 2 solution that can be obtained in polynomial time. This completes the proof. □

D. Final Remarks and Future Work

Consider a QCQP formulation of the polynomial optimization (1). There are two important factors about the SDP relaxation of this QCQP problem: (i) optimal objective value of the SDP relaxation that serves as a lower bound on the globally minimum cost of (1), and (ii) the rank of the minimum-rank solution of the SDP relaxation. It turns that the proposed sparsification techniques reduce the rank, but loosen the lower bound at the same time. On the positive side, this rank reduction facilitates the approximation of the SDP solution by a rank-1 matrix. On the negative side, the sparsification process worsens the lower bound. Hence, the sparsification introduces a trade-off between the rank of the SDP solution and its optimal objective value. Based on our empirical studies, Vertex Duplication Procedure or Edge Elimination Procedure should be repeated until a relatively low-rank, but not necessary rank-2, solution is obtained. We have applied our technique to the optimal decentralized control problem in [42] and observed in many numerical examples that the sparsification process enables the recovery of a near-global optimal control very efficiently. An important future research direction is the investigation of the abovementioned trade-off.

A feasible, global, near-global, or approximate solution of the polynomial optimization (1) may only be retrieved from a rank-1 SDP matrix. Consider an SDP relaxation with a low-rank solution. Now, three strategy could be taken to find a near-global (sub-optimal) solution of (1):

- Since $\mathbf{W}^{\text{opt}}$ has only a few undesirable (nonzero) eigenvalues, it may be converted to an approximate solution via a local search algorithm. Based on the eigenvalue decomposition, it is straightforward to design an iterative algorithm with the property that the rank of the solution does not increase at any iteration of the algorithm. This leads to a sequence of low-rank matrices, which may converge to a rank-1 solution. The obtained solution may ultimately need to be approximated by a rank-1 matrix if it is not ultimately rank-1.
- The unwanted nonzero eigenvalues of $\mathbf{W}^{\text{opt}}$ may be eliminated by means of a penalization (regularization) technique.
- Another technique is to directly approximate $\mathbf{W}^{\text{opt}}$ with a rank-1 matrix by solving a convex optimization.

We have tested the above rank-1 approximation techniques on several instances of the optimal decentralized control problem and 7000 instances of the optimal power flow problem in [42] and [20] (see [45]–[49] and the references therein for an overview of these two problems). The above ad-hoc methods worked with a very high rate of success in our simulations. However, the development of a systematic approximation technique supported by a rigorous theory is left as future work. The design of such a rounding technique makes it possible to understand how much approximation is needed to make a hard optimization problem polynomial-time solvable.

V. Conclusions

The objective of this paper is to investigate the computational complexity of an arbitrary polynomial optimization.

Three graph sparsification techniques are proposed to design a sparse quadratically-constrained quadratic program (QCQP) that is equivalent to the original polynomial optimization. To convexify the QCQP problem, a semidefinite programming (SDP) relaxation is deployed. The existence of a rank-1 matrix solution for the SDP relaxation guarantees the recovery of a global solution of the polynomial optimization. It is proved that the SDP relaxation always has a rank 1 or 2 solution. This result implies that the NP-hardness of polynomial optimization is related to going from rank 2 to rank 1 as opposed to converting a high-rank matrix to a low-rank matrix. This result helps answer one fundamental question in optimization theory: how much approximation is needed to make a hard optimization problem polynomial-time solvable? This question may be addressed by investigating the gap between the rank-2 SDP solution and its best rank-1 approximation.

References

[1] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, 2004.

[2] Y. Nesterov, A. S. Nemirovskii, and Y. Ye, *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

[3] D. P. Bertsekas, "Nonlinear programming," 1999.

[4] A. Ben-Tal and A. S. Nemirovski, *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. SIAM, 2001.

[5] K. G. Murty and S. N. Kabadi, "Some NP-complete problems in quadratic and nonlinear programming," *Mathematical programming*, vol. 39, no. 2, pp. 117–129, 1987.

[6] E. D. Klerk, "The complexity of optimizing over a simplex, hypercube or sphere: a short survey," *Central European Journal of Operations Research*, vol. 16, pp. 111–125, 2008.

[7] G. L. Nemhauser and L. A. Wolsey, *Integer and combinatorial optimization*. Wiley New York, 1988.

[8] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*. Courier Dover Publications, 1998.

[9] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, "Linear matrix inequalities in system and control theory," *Studies in Applied Mathematics, SIAM*, 1994.

[10] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *Journal of the ACM (JACM)*, vol. 42, no. 6, pp. 1115–1145, 1995.

[11] Y. Nesterov, "Semidefinite relaxation and nonconvex quadratic optimization," *Optimization Methods and Software*, vol. 9, pp. 141–160, 1998.

[12] Y. Ye, "Approximating quadratic programming with bound and quadratic constraints," *Mathematical Programming*, vol. 84, pp. 219–226, 1999.

[13] Y. Ye, "Approximating global quadratic optimization with convex quadratic constraints," *Journal of Global Optimization*, vol. 15, pp. 1–17, 1999.

[14] S. Zhang, "Quadratic maximization and semidefinite relaxation," *Mathematical Programming A*, vol. 87, pp. 453–465, 2000.

[15] S. Zhang and Y. Huang, "Complex quadratic optimization and semidefinite programming," *SIAM Journal on Optimization*, vol. 87, pp. 871–890, 2006.

[16] Z. Luo, N. Sidiropoulos, P. Tseng, and S. Zhang, "Approximation bounds for quadratic optimization with homogeneous quadratic constraints," *SIAM Journal on Optimization*, vol. 18, pp. 1–28, 2007.

[17] S. He, Z. Luo, J. Nie, and S. Zhang, "Semidefinite relaxation bounds for indefinite homogeneous quadratic optimization," *SIAM Journal on Optimization*, vol. 19, pp. 503–523, 2008.

[18] S. He, Z. Li, and S. Zhang, "Approximation algorithms for homogeneous polynomial optimization with quadratic constraints," *Mathematical Programming*, vol. 125, pp. 353–383, 2010.

[19] J. Lavaei and S. H. Low, "Zero duality gap in optimal power flow problem," *Power Systems, IEEE Transactions on*, vol. 27, no. 1, pp. 92–107, 2012.

[20] R. Madani, S. Sojoudi, and J. Lavaei, "Convex relaxation for optimal power flow problem: Mesh networks," *Technical Report (shorter version appeared in Asilomar)*, 2014. [Online]. Available: http://www.ee.columbia.edu/~lavaei/Penalized_SDP_2013.pdf

[21] M. Fazel, H. Hindi, and S. Boyd, "Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices," *American Control Conference*, vol. 3, pp. 2156–2162, 2003.

[22] B. Recht, M. Fazel, and P. Parrilo, "Guaranteed minimum rank solutions of matrix equations via nuclear norm minimization," *SIAM Review*, 2007.

[23] G. Pataki, "On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues," *Mathematics of Operations Research*, vol. 23, pp. 339–358, 1998.

[24] J. F. Sturm and S. Zhang, "On cones of nonnegative quadratic functions," *Mathematics of Operations Research*, vol. 28, pp. 246–267, 2003.

[25] Y. Huang and S. Zhang, "Complex matrix decomposition and quadratic programming," *Mathematics of Operations Research*, vol. 32, pp. 758–768, 2007.

[26] M. X. Goemans and D. P. Williamson, "Approximation algorithms for max-3-cut and other problems via complex semidefinite programming," *Journal of Computer and System Sciences*, vol. 68, pp. 422–470, 2004.

[27] J. Briet, F. Vallentin *et al.*, "Grothendieck inequalities for semidefinite programs with rank constraint," *arXiv preprint arXiv:1011.1754*, 2010.

[28] N. Alon, K. Makarychev, Y. Makarychev, and A. Naor, "Quadratic forms on graphs," *Inventiones mathematicae*, vol. 163, no. 3, pp. 499–522, 2006.

[29] M. Laurent and A. Varvitsiotis, "Computing the grothendieck constant of some graph classes," *Operations Research Letters*, vol. 39, no. 6, pp. 452–456, 2011.

[30] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisability problems using semidefinite programming," *Journal of the ACM*, vol. 42, pp. 1115–1145, 1995.

[31] J. B. Lasserre, "An explicit exact SDP relaxation for nonlinear 0-1 programs," in *Integer Programming and Combinatorial Optimization*. Springer, 2001, pp. 293–303.

[32] S. Kim and M. Kojima, "Exact solutions of some nonconvex quadratic optimization problems via SDP and SOCP relaxations," *Computational Optimization and Applications*, vol. 26, no. 2, pp. 143–154, 2003.

[33] S. Sojoudi and J. Lavaei, "On the exactness of semidefinite relaxation for nonlinear optimization over graphs: Part I," *IEEE Conference on Decision and Control*, 2013.

[34] S. Sojoudi and J. Lavaei, "On the exactness of semidefinite relaxation for nonlinear optimization over graphs: Part II," *IEEE Conference on Decision and Control*, 2013.

[35] J. Lavaei, "Zero duality gap for classical opf problem convexifies fundamental nonlinear power problems," *American Control Conference*, 2011.

[36] S. Sojoudi and J. Lavaei, "Physics of power networks makes hard optimization problems easy to solve," *IEEE Power & Energy Society General Meeting*, 2012.

[37] J. Lavaei, B. Zhang, and D. Tse, "Geometry of power flows in tree networks," *IEEE Power & Energy Society General Meeting*, 2012, (Journal version to appear in *IEEE Transactions on Power Systems*).

[38] A. Barvinok, "Problems of distance geometry and convex properties of quadartic maps," *Discrete and Computational Geometry*, vol. 12, pp. 189–202, 1995.

[39] A. Barvinok, "A remark on the rank of positive semidefinite matrices subject to affine constraints," *Discrete & Computational Geometry*, vol. 25, no. 1, pp. 23–31, 2001.

[40] W. Ai, Y. Huang, and S. Zhang, "On the low rank solutions for linear matrix inequalities," *Mathematics of Operations Research*, vol. 33, no. 4, pp. 965–975, 2008.

[41] S. Sojoudi and J. Lavaei, "Semidefinite relaxation for nonlinear optimization over graphs with application to power systems," *Preprint available at http*, 2013.

[42] G. Fazelnia, R. Madani, and J. Lavaei, "Optimal distributed control problem as a rank-constrained optimization," *Technical Report (shorter version appeared in Allerton)*, 2014. [Online]. Available: http://www.ee.columbia.edu/~lavaei/Dec_Control_2014_Report.pdf

[43] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010.

[44] R. Madani, G. Fazelnia, S. Sojoudi, and J. Lavaei, "Finding low-rank solutions of sparse linear matrix inequalities using convex optimization," *Technical Report*, 2014. [Online]. Available: http://www.ee.columbia.edu/~lavaei/LMI_Low_Rank.pdf

[45] N. Motee and A. Jadbabaie, "Optimal control of spatially distributed systems," *Automatic Control, IEEE Transactions on*, vol. 53, no. 7, pp. 1616–1629, 2008.

[46] F. Lin, M. Fardad, and M. R. Jovanovic, “Augmented lagrangian approach to design of structured optimal state feedback gains,” *IEEE Transactions on Automatic Control*, vol. 56, no. 12, pp. 2923–2929, 2011.

[47] A. Lamperski and J. C. Doyle, “Output feedback H_2 model matching for decentralized systems with delays,” *American Control Conference*, 2013.

[48] D. K. Molzahn, J. T. Holzer, B. C. Lesieutre, and C. L. DeMarco, “Implementation of a large-scale optimal power flow solver based on semidefinite programming,” to appear in *IEEE Transactions on Power Systems*, 2013.

[49] S. Bose and K. M. C. D. F. Gayme, S. H. Low, “Quadratically constrained quadratic programs on acyclic graphs with application to power flow,” *arXiv:1203.5599v1*, 2012.