Mathematical Programming (2023) 201:263–293
https://doi.org/10.1007/s10107-022-01907-3

FULL LENGTH PAPER

Series A

An approximation algorithm for indefinite mixed integer quadratic programming

Alberto Del Pia[1]

Received: 1 March 2021 / Accepted: 11 November 2022 / Published online: 2 December 2022

Abstract

In this paper, we give an algorithm that finds an ϵ-approximate solution to a mixed integer quadratic programming (MIQP) problem. The algorithm runs in polynomial time if the rank of the quadratic function and the number of integer variables are fixed. The running time of the algorithm is expected unless P = NP. In order to design this algorithm we introduce the novel concepts of spherical form MIQP and of aligned vectors, and we provide a number of results of independent interest. In particular, we give a strongly polynomial algorithm to find a symmetric decomposition of a matrix, and show a related result on simultaneous diagonalization of matrices.

Keywords Mixed integer quadratic programming · Approximation algorithm · Polynomial time · Symmetric decomposition · Simultaneous diagonalization

Mathematics Subject Classification 90C11 · 90C20 · 90C26 · 90C59

1 Introduction

Mixed Integer Quadratic Programming (MIQP) is an optimization problem where the objective function is a general quadratic function, the constraints are linear inequalities, and some of the variables are required to be integers. Formally, given a symmetric matrix $H \in \mathbb{Q}^{n\times n}$, a matrix $W \in \mathbb{Q}^{m\times n}$, vectors $h \in \mathbb{Q}^n$, $w \in \mathbb{Q}^m$, and $p \in \{0, 1, \dots, n\}$, we seek a vector $x \in \mathbb{R}^n$ that attains

$$\begin{aligned} \min \quad & x^\mathsf{T} H x + h^\mathsf{T} x \\ \text{s.t.} \quad & Wx \le w \\ & x \in \mathbb{Z}^p \times \mathbb{R}^{n-p}. \end{aligned} \tag{MIQP}$$

✉ Alberto Del Pia
delpia@wisc.edu

[1] Department of Industrial and Systems Engineering and Wisconsin Institute for Discovery, University of Wisconsin-Madison, Madison, WI, USA

Many important applications can be modeled as MIQPs, in areas such as operations research, engineering, computer science, physics, biology, finance, economics, and artificial intelligence. MIQP reduces to *Mixed Integer Linear Programming* (MILP) when H is a zero matrix, and to *Quadratic Programming* (QP) if $p = 0$. Moreover, MIQP is a prototypical *Mixed Integer Nonlinear Programming* (MINLP) problem, as it captures the critical elements of those models, but in the simplest possible way, making it the natural first step to construct efficient algorithms for MINLP.

The decision version of MIQP lies in the complexity class NP [10]. Furthermore, MIQP is strongly NP-hard [15], and remains NP-hard even if H has rank one and $p = 0$ [28]. This implies the lack of efficient algorithms for solving this class of optimization problems in its full generality.

The main result of this paper is an approximation algorithm for MIQP. In order to state our result, we first give the definition of ϵ-approximate solution. Consider an instance of (MIQP), and assume that it has an optimal solution x^*. Let $f(x)$ denote the objective function, and let $f_{\max}$ be the maximum value of $f(x)$ on the feasible region. For $\epsilon \in [0, 1]$, we say that a feasible point $x^\diamond$ is an ϵ*-approximate solution* if

$$f(x^\diamond) - f(x^*) \leq \epsilon \cdot (f_{\max} - f(x^*)).$$

Note that only optimal solutions are 0-approximate solutions, while any feasible point is a 1-approximate solution. The definition of ϵ-approximate solution has some useful invariance properties which make it a natural choice in this setting. For instance, it is preserved under dilation and translation of the objective function, and it is insensitive to affine transformations of the objective function and of the feasible region, like for example changes of basis. Our definition of approximation has been used in earlier works, and we refer to [1, 22, 27, 31] for more details. We can now state our main result.

Theorem 1 *For every $\epsilon \in (0, 1]$, there is an algorithm that finds an ϵ-approximate solution to a bounded (MIQP), if it exists. The running time of the algorithm is polynomial in the size of the input and in $1/\epsilon$, provided that the rank k of the matrix H and the number of integer variables p are fixed numbers.*

This is the first known polynomial time approximation algorithm for MIQP with k and p fixed. In particular, note that the dimension n of the problem is not required to be fixed. The running time of the algorithm exhibits a polynomial dependence on the size of the instance and on $1/\epsilon$, and an exponential dependence on k and on p. It is known that this dependence is expected unless P=NP, and we refer the reader to the discussion below the statement of Theorem 1 in [8].

One might wonder if the boundedness assumption can be relaxed in Theorem 1, with the understanding that, if the input MIQP is unbounded, the algorithm should return at least a statement that the instance is unbounded. The next theorem implies that the boundedness assumption cannot be removed unless P=NP.

Theorem 2 *Determining whether (MIQP) is unbounded is NP-complete, even if the rank k of the matrix H equals three and the number p of integer variables is zero.*

Proof From Theorem 4 in [10], the decision problem in the statement is in NP, thus we only need to show the NP-hardness. In Sects. 2 and 3 in [28], the authors present a QP of the form

$$\min\{x_1 - x_2^2 : Wx \leq w,\ x \in \mathbb{R}^n\} \tag{1}$$

with nonnegative optimum objective value, and for which it is NP-hard to determine if the optimum value is zero. Since every bounded QP has an optimal solution of polynomial size [29], there is a number ϕ which is polynomial in the size of the input QP (1) for which the optimum objective value is either zero or strictly larger than $2^{-\phi}$.

Consider now the QP

$$\min\{x_1x_{n+1} - x_2^2 - 2^{-\phi}x_{n+1}^2 : (W \mid -w)x \leq 0,\ x_{n+1} \geq 1,\ x \in \mathbb{R}^{n+1}\}. \tag{2}$$

Notice that the rank of the objective function is three. Thus, to conclude the proof of the theorem we only need to show that (2) is unbounded if and only if the optimum value of (1) is zero.

Assume that the optimum value of (1) is zero. Then there is a point $\bar{x} \in \mathbb{R}^n$ with $W\bar{x} \leq w$ and $\bar{x}_1 - \bar{x}_2^2 = 0$. Consider now the set of vectors in $\mathbb{R}^{n+1}$ given by $(\lambda\bar{x}, \lambda)$, for $\lambda \geq 1$. Note that all these vectors are feasible to (2). Furthermore, the objective value of $(\lambda\bar{x}, \lambda)$ is $\lambda^2(\bar{x}_1 - \bar{x}_2^2 - 2^{-\phi}) = -\lambda^2 2^{-\phi}$ which goes to $-\infty$ as $\lambda \to \infty$. Therefore, (2) is unbounded.

Next, assume that the optimum value of (1) is positive, therefore strictly larger than $2^{-\phi}$. Consider a vector feasible to (2) and note that it can be written as $(\bar{\lambda}\bar{x}, \bar{\lambda})$, where $\bar{\lambda} \geq 1$ and $\bar{x}$ satisfies $W\bar{x} \leq w$. The objective value of $(\bar{\lambda}\bar{x}, \bar{\lambda})$ is $\bar{\lambda}^2(\bar{x}_1 - \bar{x}_2^2 - 2^{-\phi})$. Since $\bar{x}$ is feasible to (1), we have $\bar{x}_1 - \bar{x}_2^2 > 2^{-\phi}$, thus the objective value of $(\bar{\lambda}\bar{x}, \bar{\lambda})$ is positive. In particular, (2) is bounded. □

In particular, Theorem 2 strengthens the result by Murty and Kabadi [26] that deciding whether a QP is bounded or not is NP-hard.

1.1 Literature review

In this section, we review the known exact and approximation algorithms for MIQP with a polynomial running time.

MIQP admits a polynomial time approximation algorithm if the dimension n is fixed [6]. MIQP is polynomially solvable if the dimension n is fixed and the objective is convex [21] or concave [3, 4, 18]. If the objective is concave with a fixed number of negative eigenvalues and the number p of integer variables is fixed, there is a polynomial time approximation algorithm [8].

Next, we survey *Integer Quadratic Programming* (IQP), which is the special case of MIQP where all variables are integer, i.e., $p = n$. IQP is solvable in polynomial time in dimension one and two [11]. Furthermore, there is a polynomial time approximation algorithm if the dimension is fixed and the objective is homogeneous with at most one positive or negative eigenvalue [19]. If the objective function is separable and convex, and the constraint matrix W is TU, then IQP can be solved in polynomial time [20].

IQP admits a polynomial time approximation algorithm if the objective is separable and concave, with a fixed number of negative eigenvalues, and the largest absolute value of the subdeterminants of the constraint matrix is bounded by two [9]. Other IQP tractability results under specific structural restrictions can be found in [13, 24].

Finally, we discuss *Quadratic Programming* (QP), the special case of MIQP where all variables are continuous, i.e., $p = 0$. QP can be solved in polynomial time if the dimension is fixed [10, 29]. Furthermore, QP admits a polynomial time approximation algorithm if the number of negative eigenvalues of H is fixed [30], and it admits a weaker polynomial time approximation algorithm in general [32]. If the objective is convex, then QP can be solved in polynomial time [23].

1.2 Overview of the results and organization of the paper

Our approximation algorithm is based on the novel concepts of *spherical form MIQP* and of *aligned vectors*. These two notions significantly enhance the available mathematical toolkit for the design and analysis of algorithms for MIQP, and therefore their importance is not limited to this work.

Sections 2 and 3 are devoted to finding a change of basis that transforms a MIQP in spherical form. In a *spherical form MIQP* the objective is separable and the polyhedron has a "spherical appearance". Moreover, the set $\mathbb{Z}^p \times \mathbb{R}^{n-p}$ is replaced by a set of the form $\Lambda + \operatorname{span}(\Lambda)^\perp$, for a lattice Λ of rank p. The formal definition is given in Sect. 3. In order to obtain this change of basis we develop a number of results of independent interest.

Since a spherical form MIQP has a separable objective function, in particular we need to find an invertible matrix L and a diagonal matrix D such that $H = LDL^\mathsf{T}$. In Sect. 2 we focus on this simpler task and, in Theorem 3 and Corollary 1, we present a *symmetric decomposition algorithm* that constructs such matrices L, D in strongly polynomial time. This is the first known polynomial time algorithm for this problem.

In Sect. 3, we build on this algorithm and obtain, in Proposition 1, a rational version of theorems on simultaneous diagonalization of matrices. In particular, we show that we can find in polynomial time an invertible matrix L that at the same time diagonalizes a given matrix H, and provides the shape of an ellipsoid that approximates a given polytope within a factor depending only on the dimension. This result is the main building block that allows us to obtain, in Proposition 2, a polynomial time algorithm to transform a MIQP in spherical form.

In Sect. 4 we introduce the concept of *aligned vectors* for a spherical form MIQP. In particular, they are two feasible vectors that are "far" in the direction where the objective is "most curved" and "almost aligned" in all other directions. Furthermore, their midpoint is feasible as well. We then show, in Proposition 3, that if a spherical form MIQP has two aligned vectors, then it is possible to find an ϵ-approximate solution by solving a number of MILPs. This number is polynomial in $1/\epsilon$ if both k and p are fixed in the original (MIQP).

In Sect. 5 we focus on the problem of deciding whether a spherical form MIQP has two aligned vectors or not. In Proposition 5 we give a polynomial time algorithm that either finds two aligned vectors, or finds a vector $v \in \operatorname{span}(\Lambda)$ along which the polyhedron is “flat”. The vector v allows us to decompose the problem in a number of MIQPs with fewer integer variables. Furthermore, this number depends only on k and p, and thus is a constant if both k and p are fixed.

In Sect. 6 we then present our approximation algorithm for MIQP and provide a proof of Theorem 1. The algorithm first uses Proposition 2 to find a change of basis that transforms the input MIQP in spherical form. Then, it employs Proposition 5 and it either finds two aligned vectors, or finds a vector $v \in \text{span}(\Lambda)$ along which the polyhedron is “flat”. In the first case, we use Proposition 3 to find an ϵ-approximate solution. In the second case, the input MIQP is decomposed into a constant number of instances with fewer integer variable, and the algorithm is recursively applied to these instances. At the end of the execution, the algorithm returns the best solution found, and we show that it is an ϵ-approximate solution to the input MIQP.

In this paper, we will be using several concepts from computational complexity. Recall that a *strongly polynomial algorithm* is a polynomial space algorithm in the Turing model and a polynomial time algorithm in the arithmetic model. The definition of strong polynomiality mixes the Turing model and the arithmetic model of computation. Throughout the paper, unless we state a different model, we mean the Turing model. For more details on time and space complexity we refer the reader to [17]. In particular, we recall that a strongly polynomial algorithm is also a polynomial time algorithm.

2 A strongly polynomial algorithm for symmetric decomposition

Given a rational symmetric $n \times n$ matrix H, a *symmetric decomposition* of H is a decomposition of the form $BHB^{\mathsf{T}} = D$, where B is an $n \times n$ nonsingular matrix and D is an $n \times n$ diagonal matrix. The goal of this section is to give an algorithm that constructs a symmetric decomposition of any rational symmetric matrix H with two fundamental properties: (i) the algorithm is strongly polynomial, (ii) the Frobenius norms of B and B^{-1} are upper bounded by an integer of size polynomial in n. To the best of our knowledge, our algorithm is the first known polynomial time algorithm that finds a symmetric decomposition of any rational symmetric matrix. Note that the spectral decomposition, the Schur decomposition, and Takagi’s factorization yield a symmetric decomposition of a rational symmetric matrix. However, none of these decompositions can be performed in polynomial space since the resulting matrices generally contain irrational elements. Other related matrix decompositions are the Cholesky decomposition and the LDL^{T} decomposition, but are not applicable to indefinite matrices. For more details on matrix decompositions we refer the reader to [16].

By introducing pivoting operations that perform symmetric additions of rows and columns, as well as symmetric interchanges, Dax and Kaniel [5] describe an algorithm that constructs a symmetric decomposition of any symmetric $n \times n$ matrix H. Their algorithm performs a number of arithmetic operations that is polynomial in n, thus it is a polynomial time algorithm in the arithmetic model. However, it is unknown if it is a polynomial time algorithm or a polynomial space algorithm.

In this section, we present a strongly polynomial version of Dax and Kaniel's algorithm. Therefore, for our version of the algorithm, we show that all numbers stored during the execution of the algorithm have size that is polynomial in the size of the input matrix H. This in particular implies that the output matrices B and D have polynomial size. The proof builds on the technique introduced by Edmonds to perform Gaussian elimination in strongly polynomial time [12], but it is more involved due to the "complete pivoting" performed at each iteration. In particular, while in Gaussian elimination every number stored during the algorithm is a ratio of subdeterminants of the original matrix, every number stored in our version Dax and Kaniel's algorithm at iteration k is shown to be a ratio of subdeterminants of the matrix obtained from H by performing only the first k pivoting operations.

Another fundamental property of our symmetric decomposition algorithm is that the Frobenius norms of B and B^{-1} are upper bounded by an integer of size polynomial in n. In particular, this integer depends only on n and not on the input matrix. This property will be fundamental in the next sections of the paper, where the symmetric decomposition algorithm will be used to obtain a change of basis for our MIQP. Recall that the *Frobenius norm* of an $m \times n$ matrix A is defined by $\|A\|_F := \sqrt{\sum_{i=1}^m \sum_{j=1}^n A_{ij}^2}$.

Therefore, the purpose of this section is to prove the following result.

Theorem 3 *Let H be a rational symmetric $n \times n$ matrix. There is a strongly polynomial algorithm that finds matrices B, D such that $BHB^\mathsf{T} = D$ is a symmetric decomposition of H. Furthermore, $\|B\|_F$ and $\|B^{-1}\|_F$ are upper bounded by $(5n)^{n/2}$.*

If we set $L := B^{-1}$ in Theorem 3, we obtain $H = LDL^\mathsf{T}$. Since the inverse can be computed in strongly polynomial time [12], also this decomposition can be obtained in strongly polynmomial time.

Corollary 1 *Let H be a rational symmetric $n \times n$ matrix. There is a strongly polynomial algorithm that finds an invertible $n \times n$ matrix L and an $n \times n$ diagonal matrix D such that $H = LDL^\mathsf{T}$. Furthermore, $\|L\|_F$ and $\|L^{-1}\|_F$ are upper bounded by $(5n)^{n/2}$.*

Corollary 1 then provides a strongly polynomial algorithm to compute a change of basis that transforms a general (MIQP) in a separable form. Namely, compute the decomposition $H = LDL^\mathsf{T}$ and set $y := L^\mathsf{T}x$. In particular, our approach can be substituted to the techniques used in [7, 8, 19, 31] to transform the original QP or MIQP in a separable form.

In the remainder of this section we only consider matrices that are $n \times n$, thus we avoid repeating it throughout the section.

2.1 Description of the symmetric decomposition algorithm

In this section, we describe the symmetric decomposition algorithm that we analyze. It is the version of Dax and Kaniel's algorithm where the parameter γ is always chosen in ± 1.

Let H be the rational symmetric $n \times n$ matrix given in the input. Let $H^{(0)} := H$, and for every $k = 1, \dots, n-1$, we denote by $H^{(k)}$ the $n \times n$ matrix obtained after

k iterations of the algorithm. The matrix $H^{(k)}$ is symmetric and all the off-diagonal elements in the first k rows and columns equal zero. In particular, $H^{(n-1)}$ is a diagonal matrix and coincides with the matrix D in the output.

For any $k = 1, \dots, n-1$, we now describe the iteration k of the symmetric decomposition algorithm, where the matrix $H^{(k)}$ is obtained from $H^{(k-1)}$. The iteration is subdivided into two stages, called "pivoting" and "elimination".

Pivoting. The goal of the pivoting stage is to ensure that the *pivotal element,* which is the element in the (k, k) position, is one with largest absolute value among rows and columns $k, \dots, n$. Let s and r be indices such that $|H_{sr}^{(k-1)}| = \max_{i,j\in\{k,\dots,n\}} |H_{ij}^{(k-1)}|$. Since $H_{sr}^{(k-1)} = H_{rs}^{(k-1)}$ we can assume without loss of generality that $s \le r$. Let $\tilde{H}$ be the symmetric $n \times n$ matrix obtained from $H^{(k-1)}$ by interchanging rows s and k, and interchanging columns s and k. If $s = r$, then $H_{sr}^{(k-1)} = \tilde{H}_{kk}$. In this case, we have achieved our goal and the pivoting is terminated. Thus, we now assume $s < r$. This implies that $H_{sr}^{(k-1)} = \tilde{H}_{rk}$. We define

$$\gamma := \begin{cases} +1 & \text{if } \tilde{H}_{rk}(\tilde{H}_{kk} + \tilde{H}_{rr}) \ge 0 \\ -1 & \text{if } \tilde{H}_{rk}(\tilde{H}_{kk} + \tilde{H}_{rr}) < 0, \end{cases}$$

and we let $\tilde{\tilde{H}}$ be the symmetric $n \times n$ matrix obtained from $\tilde{H}$ by adding row r multiplied by γ to row k, and adding column r multiplied by γ to column k. It is simple to check that the new (k, k) element is the one with largest absolute value among rows and columns $k, \dots, n$, i.e., $|\tilde{\tilde{H}}_{kk}| = \max_{i,j\in\{k,\dots,n\}} |\tilde{\tilde{H}}_{ij}|$.

Pivoting can be achieved via matrix multiplication. We define the matrix $\tilde{P}_k$ which interchanges rows s and k, thus it is the permutation matrix obtained from the identity matrix by interchanging rows s and k (note that, if $s = k$, then $\tilde{P}_k$ is the identity matrix). The matrix $\tilde{\tilde{P}}_k$ adds (if necessary) the row r multiplied by γ to row k, therefore, it is the identity matrix if $s = r$, or it is obtained from the identity matrix by replacing the zero element in the (k, r) position with the scalar γ. The matrix $\tilde{H}$ can then be written as $\tilde{H} = \tilde{P}_k H^{(k-1)} \tilde{P}_k^{\mathsf{T}}$, while the matrix $\tilde{\tilde{H}}$ is the product $\tilde{\tilde{H}} = \tilde{\tilde{P}}_k \tilde{H} \tilde{\tilde{P}}_k^{\mathsf{T}} = P_k H^{(k-1)} P_k^{\mathsf{T}}$, where $P_k := \tilde{\tilde{P}}_k \tilde{P}_k$.

Elimination. The goal of this stage is to obtain zeros in the off-diagonal elements of row and column k. We first perform row elimination and then column elimination. The row elimination is done as in Gaussian elimination: For each $i = k+1, \dots, n$, add row k multiplied by $-\tilde{\tilde{H}}_{ik}/\tilde{\tilde{H}}_{kk}$ to row i. The column elimination is done symmetrically: for each $j = k+1, \dots, n$, add column k multiplied by $-\tilde{\tilde{H}}_{kj}/\tilde{\tilde{H}}_{kk}$ to column j.

Row elimination is performed by multiplying on the left by the matrix $(I - E_k)$, where I denotes the $n \times n$ identity matrix and the elements of E_k are given by

$$(E_k)_{ik} := \tilde{\tilde{H}}_{ik}/\tilde{\tilde{H}}_{kk} \qquad i = k+1, \dots, n, \tag{3}$$

and all the other elements are zeros. Symmetrically, column elimination is performed by multiplying on the right by the matrix $(I - E_k)^{\mathsf{T}}$. Therefore, the matrix $H^{(k)}$ is obtained from $H^{(k-1)}$ via the matrix product

$$H^{(k)} := [(I - E_k)P_k]H^{(k-1)}[(I - E_k)P_k]^\mathsf{T}. \tag{4}$$

This completes the description of the iteration k of the symmetric decomposition algorithm. At the end of iteration $n - 1$ the algorithm returns the diagonal matrix $D := H^{(n-1)}$ and the nonsingular matrix

$$B := (I - E_{n-1})P_{n-1} \cdots (I - E_1)P_1.$$

It follows directly from the description of the algorithm that the algorithm is correct, i.e., $BHB^\mathsf{T} = D$ is a symmetric decomposition of H.

2.2 Analysis of the algorithm

In this section, we prove the first part of Theorem 3. Namely, we show that the symmetric decomposition algorithm presented in Sect. 2.1 runs in strongly polynomial time. Clearly, the number of arithmetic operations performed is polynomial in n. Therefore, we only need to show that the size of each matrix constructed during the execution is polynomial in the size of H. For matrices $\tilde{P}_k$, $\tilde{\tilde{P}}_k$, P_k, for $k = 1, \dots, n - 1$, this follows directly from their definition. In fact, we only need to show that each matrix $H^{(k)}$, for $k = 1, \dots, n - 1$, has size polynomial in the size of H. Indeed, once this is proven, we obtain that also E_k and the returned matrix B have size polynomial in the size of H.

Thus, we now focus our attention on the matrix $H^{(k)}$. From the equality (4) we deduce that

$$H^{(k)} = B^{(k)} H {B^{(k)}}^\mathsf{T},$$

where

$$B^{(k)} := (I - E_k)P_k(I - E_{k-1})P_{k-1} \cdots (I - E_1)P_1.$$

As noticed on page 224 in [5], it is simple to verify that for every $t, j \in \{1, \dots, n-1\}$ with $t < j$, we have $E_t P_j = E_t$. This in turn implies that for every $t, j \in \{1, \dots, n-1\}$ with $t < j$, we have

$$P_j(I - P_{j-1}P_{j-2} \cdots P_{t+1}E_t) = (I - P_jP_{j-1} \cdots P_{t+1}E_t)P_j,$$

which allows us to write $B^{(k)}$ in the form

$$B^{(k)} = (I - E_k)(I - P_kE_{k-1}) \cdots (I - P_k \cdots P_2E_1)P_k \cdots P_1. \tag{5}$$

Therefore $H^{(k)}$ can be written as

$$H^{(k)} = E^{(k)} G^{(k)} {E^{(k)}}^\mathsf{T}, \tag{6}$$

where

$$
\begin{aligned}
G^{(k)} &:= (P_k \cdots P_1) H (P_k \cdots P_1)^{\mathsf{T}}, \\
E^{(k)} &:= (I - E_k)(I - P_k E_{k-1}) \cdots (I - P_k \cdots P_2 E_1).
\end{aligned}
$$

In the next lemma, we analyze the matrices $P_k P_{k-1} \cdots P_{t+1} E_t$ in the definition of $E^{(k)}$. The second part of the statement will only be used later in Sect. 2.3.

Lemma 1 *For each $t \in \{1, \ldots, k\}$, the matrix $P_k P_{k-1} \cdots P_{t+1} E_t$ can have nonzeros only in positions $(t+1, t), \ldots, (n, t)$. Furthermore, the elements in rows $t+1, \ldots, k$ are bounded by two in absolute value, while the elements in rows $k+1, \ldots, n$ are bounded by one in absolute value.*

Proof We show this lemma by induction on $k - t$. In the base case we have $k - t = 0$, thus we are considering matrix E_t. By definition, E_t can have nonzeros only in positions $(t+1, t), \ldots, (n, t)$, and from (3) all nonzeros are are bounded by one in absolute value.

For the inductive step we assume $k - t \geq 1$ and consider the matrix $P_k(P_{k-1} \cdots P_{t+1} E_t)$. By induction, $P_{k-1} \cdots P_{t+1} E_t$ can have nonzeros only in positions $(t+1, t), \ldots, (n, t)$. Furthermore, the elements in rows $t+1, \ldots, k-1$ are bounded by two in absolute value, while the elements in rows $k, \ldots, n$ are bounded by one in absolute value. We have $P_k = \tilde{\tilde{P}}_k \tilde{P}_k$, where the matrix $\tilde{P}_k$ interchanges two rows in $\{k, \ldots, n\}$, and the matrix $\tilde{\tilde{P}}_k$ adds or subtracts (if necessary) a row in $\{k+1, \ldots, n\}$ to row k. Since $k \geq t+1$, the matrix $P_k(P_{k-1} \cdots P_{t+1} E_t)$ can have nonzeros only in positions $(t+1, t), \ldots, (n, t)$. The elements in rows $t+1, \ldots, k-1$ are left unchanged, thus they are bounded by two in absolute value. The element in row k is now bounded by two in absolute value, while the elements in rows $k+1, \ldots, n$ remain bounded by one in absolute value. □

Next, we use Lemma 1 to discuss the effect of multiplying a matrix on the left by $E^{(k)}$. Note that a multiplication of this type is performed in (6).

Lemma 2 *Multiplying a matrix on the left by $E^{(k)}$ results in a sequence of elementary row operations in which a multiple of a row $t \in \{1, \ldots, k\}$ is added to a row in $\{t+1, \ldots, n\}$.*

Proof Due to the definition of $E^{(k)}$, it suffices to show that multiplying a matrix on the left by $(I - P_k \cdots P_{t+1} E_t)$, for $t \in \{1, \ldots, k\}$, results in a sequence of elementary row operations in which a multiple of row t is added to a row in $\{t+1, \ldots, n\}$.

From Lemma 1, the matrix $P_k \cdots P_{t+1} E_t$ can have nonzeros only in positions $(t+1, t), \ldots, (n, t)$. Hence, the multiplication on the left by matrix $(I - P_k \cdots P_{t+1} E_t)$ preserves the first t rows, and each subsequent row is obtained by adding a multiple of row t to the original row. □

We are finally ready to show, in the next claim, that each matrix $H^{(k)}$ has size polynomial in the size of H. This concludes the proof that our symmetric decomposition algorithm runs in strongly polynomial time.

Claim 1 *For each $k \in \{1, \ldots, n-1\}$, the size of $H^{(k)}$ is polynomial in the size of H.*

Proof Let $G_k^{(k)}$ denote the $k \times k$ submatrix of $G^{(k)}$ determined by the first k rows and columns, and let $G_{k|ij}^{(k)}$, for $i, j \in \{k+1, \ldots, n\}$, denote the $(k+1) \times (k+1)$ submatrix of $G^{(k)}$ determined by rows $\{1, \ldots, k, i\}$ and columns $\{1, \ldots, k, j\}$.

It suffices to show that for every $k \in \{1, \ldots, n-1\}$ and for every $i, j \in \{k+1, \ldots, n\}$, we have

$$H_{ij}^{(k)} = \det(G_{k|ij}^{(k)}) / \det(G_k^{(k)}). \tag{7}$$

In fact, the definition of $G^{(k)}$ implies that its size is polynomial in the size of H. Therefore, also $\det(G_{k|ij}^{(k)})$ and $\det(G_k^{(k)})$ have size polynomial in the size of H, and so does each element of $H^{(k)}$ due to (7). Therefore, in the remainder of the proof we show (7).

Consider the product $E^{(k)}G^{(k)}$ in (6). From Lemma 2, the resulting matrix is obtained from $G^{(k)}$ via a sequence of elementary row operations. Among all these elementary row operations, only a subset modify the first k rows of the matrix $G^{(k)}$. From Lemma 2, in each of these elementary row operations, a multiple of a row $t \in \{1, \ldots, k-1\}$ is added to a row in $\{t+1, \ldots, k\}$. Similarly, among all the elementary column operations performed by ${E^{(k)}}^{\mathsf{T}}$, only a subset modify the first k columns of the matrix $G^{(k)}$. In each of these elementary column operations, a multiple of a column $t \in \{1, \ldots, k-1\}$ is added to a column in $\{t+1, \ldots, k\}$. We perform these subsets of elementary row and column operations to the matrix $G_k^{(k)}$. From (6), the resulting matrix is precisely the submatrix of $H^{(k)}$ given by the first k rows and columns, hence it is diagonal with elements $H_{11}^{(k)}, \ldots, H_{kk}^{(k)}$ in the diagonal. Note that each elementary operation considered preserves the determinant of $G_k^{(k)}$. Thus,

$$\det(G_k^{(k)}) = H_{11}^{(k)} \cdots H_{kk}^{(k)}. \tag{8}$$

A similar argument can be applied to the matrix $G_{k|ij}^{(k)}$. Among all the elementary row operations performed by $E^{(k)}$, only a subset modify rows $\{1, \ldots, k, i\}$ of the matrix $G^{(k)}$. From Lemma 2, in each of these elementary row operations, a multiple of a row $t \in \{1, \ldots, k\}$ is added to a row in $\{t+1, \ldots, k, i\}$. Similarly, among all the elementary column operations performed by ${E^{(k)}}^{\mathsf{T}}$, only a subset modify columns $\{1, \ldots, k, j\}$ of the matrix $G^{(k)}$. In each of these elementary column operations, a multiple of a column $t \in \{1, \ldots, k\}$ is added to a column in $\{t+1, \ldots, k, j\}$. We perform this subset of elementary operations to the matrix $G_{k|ij}^{(k)}$. From (6), the resulting matrix is precisely the submatrix of $H^{(k)}$ determined by rows $\{1, \ldots, k, i\}$ and columns $\{1, \ldots, k, j\}$. Hence it is diagonal with elements $H_{11}^{(k)}, \ldots, H_{kk}^{(k)}, H_{ij}^{(k)}$ in the diagonal. Each elementary operation considered preserves the determinant of $G_{k|ij}^{(k)}$. Thus, we have

$$\det(G_{k|ij}^{(k)}) = H_{11}^{(k)} \cdots H_{kk}^{(k)} \cdot H_{ij}^{(k)}.$$

Dividing the latter equation by Eq. (8), we obtain (7). □

2.3 Frobenius norm of B and B^{-1}

In this section, we prove the second part of Theorem 3. Namely, we show that for the matrix B returned by the algorithm described in Sect. 2.1, both its Frobenius norm and the Frobenius norm of its inverse are upper bounded by an integer of size polynomial in n. We will use the fact that the Frobenius norm is *submultiplicative*, i.e., for matrices A, A' we have $\|AA'\|_F \le \|A\|_F \cdot \|A'\|_F$.

Claim 2 *The Frobenius norm of matrices B and B^{-1} is upper bounded by $(5n)^{n/2}$.*

Proof From (5), we can write $B = B^{(n-1)}$ as the product $B = EP$, where

$$E := E^{(n-1)} = (I - E_{n-1})(I - P_{n-1}E_{n-2}) \cdots (I - P_{n-1} \cdots P_2 E_1),$$
$$P := P_{n-1}P_{n-2} \cdots P_1.$$

In order to bound the Frobenius norm of B and B^{-1}, we bound separately the Frobenius norm of P, P^{-1}, E, and E^{-1}.

Norm of P. Recall that each matrix P_k is the product of two matrices $P_k = \tilde{\tilde{P}}_k \tilde{P}_k$, where the matrix $\tilde{P}_k$ interchanges row s and k, where $s \ge k$, and the matrix $\tilde{\tilde{P}}_k$ adds (if necessary) row r multiplied by γ to row k, where $r > k$. Therefore, for each $k = n-1, \dots, 1$, in the matrix $P_k \cdots P_1$, the last $n-k$ rows are permuted rows of the identity matrix, while each of the first k rows has at most two nonzero elements, each one being ±1. We obtain

$$\|P\|_F = \|P_{n-1} \cdots P_1\|_F \le \sqrt{2n-1}.$$

Norm of P^{-1}. Each matrix P_k^{-1} is the product $P_k^{-1} = \tilde{P}_k^{-1} \tilde{\tilde{P}}_k^{-1}$, where the matrix $\tilde{\tilde{P}}_k^{-1}$ adds (if necessary) row r multiplied by $-\gamma$ to row k, where $r > k$, and the matrix $\tilde{P}_k^{-1}$ interchanges row s and k, where $s \ge k$. Therefore, for each $k = n-1, \dots, 1$, in the matrix $P_k^{-1} P_{k+1}^{-1} \cdots P_{n-1}^{-1}$, the first $k-1$ rows coincide with the first $k-1$ rows of the identity matrix, and the remaining rows are a permutation of the rows from k to n of an upper triangular matrix with elements in $0, \pm 1$. We obtain

$$\|P^{-1}\|_F = \|P_1^{-1} P_2^{-1} \cdots P_{n-1}^{-1}\|_F \le \sqrt{(n^2+n)/2}.$$

Norm of E. From Lemma 1 with $k = n-1$, for each $t \in \{1, \dots, n-1\}$, the matrix $P_{n-1}P_{n-2} \cdots P_{t+1}E_t$ can have nonzeros only in positions $(t+1, t), \dots, (n, t)$. Furthermore, the elements in rows $t+1, \dots, n-1$ are bounded by two in absolute value, while the element in the last row is bounded by one in absolute value. Thus, we obtain

$$\|I - P_{n-1}P_{n-2} \cdots P_{t+1}E_t\|_F \le \sqrt{(n+1) + 4(n-t-1)} \le \sqrt{5(n-1)}.$$

Hence

$$\begin{aligned}\|E\|_F &\le \|I - E_{n-1}\|_F \cdot \|I - P_{n-1}E_{n-2}\|_F \cdots \|I - P_{n-1}\cdots P_2 E_1\|_F\\ &\le \sqrt{(5(n-1))^{n-1}}.\end{aligned}$$

Norm of E^{-1}. Once again, from Lemma 1 with $k = n - 1$, we know that for each $t \in \{1, \dots, n-1\}$, the matrix $P_{n-1}P_{n-2}\cdots P_{t+1}E_t$ can have nonzeros only in positions $(t+1, t), \dots, (n, t)$. This fact allows us to write E^{-1} as

$$\begin{aligned}E^{-1} &= (I - P_{n-1}\cdots P_2E_1)^{-1}\cdots(I - P_{n-1}E_{n-2})^{-1}(I - E_{n-1})^{-1}\\ &= (I + P_{n-1}\cdots P_2E_1)\cdots(I + P_{n-1}E_{n-2})(I + E_{n-1})\\ &= I + P_{n-1}\cdots P_2E_1 + \cdots + P_{n-1}E_{n-2} + E_{n-1}.\end{aligned}$$

In particular, the matrix E^{-1} is *unit lower triangular,* i.e., lower triangular with all elements on the main diagonal equal to one. The second part of Lemma 1 then implies that the elements in rows $1, \dots, n-1$ are bounded by two in absolute value, while the elements in the last row are bounded by one in absolute value. We obtain

$$\|E^{-1}\|_F \le \sqrt{(2n-1) + 4(n^2 - 3n + 2)/2} = \sqrt{2n^2 - 4n + 3}.$$

Norm of B and B^{-1}. Using the obtained bounds on the Frobenius norm of P, P^{-1}, E, E^{-1}, we derive

$$\begin{aligned}\|B\|_F &= \|EP\|_F \le \|E\|_F\,\|P\|_F \le \sqrt{(5(n-1))^{n-1}(2n-1)},\\ \|B^{-1}\|_F &= \|P^{-1}E^{-1}\|_F \le \|P^{-1}\|_F\,\|E^{-1}\|_F \le \sqrt{(n^2+n)(2n^2-4n+3)/2}.\end{aligned}$$

It can be checked that $(5n)^{n/2}$ is larger than both upper bounds for any n. Therefore, the claim follows. □

While the bound on the Frobenius norm of matrices B and B^{-1} in Claim 2 is sufficient for our task, we remark that a better bound can be obtained by providing a better bound on $\|E\|_F$. This can be done by bounding the largest absolute value of an element in E, instead of using the fact that the Frobenius norm is submultiplicative.

3 Simultaneous diagonalization and spherical form MIQP

In this section, a fundamental role is played by the spherical form MIQP. To formally define this problem, we now briefly recall the notion of lattice, and introduce some notation.

Given linearly independent vectors $b^1, \dots, b^p$ in $\mathbb{R}^d$, the *lattice* generated by $b^1, \dots, b^p$ is the set $\Lambda := \left\{\sum_{i=1}^p v_i b^i : v_i \in \mathbb{Z}\ \forall i = 1, \dots, p\right\}$ of integer linear combinations of the vectors b^i, for $i = 1, \dots, p$. The *rank* of Λ is p and the *dimension* of Λ is d. If $p = d$, then Λ is said to be a *full rank lattice*. Note that, in this paper, we will consider mainly lattices that are not full rank. The vectors $b^1, \dots, b^p$ are called a *lattice basis*. Given a vector $a \in \mathbb{R}^d$ and a nonnegative scalar r, we denote by $\mathcal{B}(a, r)$ the *closed ball* with center a and radius r. Formally,

$$\mathcal{B}(a, r) := \{x \in \mathbb{R}^d : \|x - a\| \le r\}.$$

Note that, throughout the paper, we use the *euclidian vector norm* defined as $\|x\| := \sqrt{x^\mathsf{T} x}$. Given vectors $x^1, \dots, x^t$, we denote by $(x^1, \dots, x^t)$ the vector $({x^1}^\mathsf{T}, \dots, {x^t}^\mathsf{T})^\mathsf{T}$. The orthogonal complement of a linear space $\mathcal{L}$ is denoted by $\mathcal{L}^\perp$.

We are now in a position to give the formal definition of a spherical form MIQP. A *spherical form MIQP* is an optimization problem of the form

$$\begin{aligned} \min \quad & y^\mathsf{T} D y + c^\mathsf{T} y + l^\mathsf{T} z \\ \text{s.t.} \quad & (y, z) \in \mathcal{P} \\ & y \in \Lambda + \text{span}(\Lambda)^\perp, \ z \in \mathbb{R}^{n-d}. \end{aligned} \tag{S-MIQP}$$

In this formulation, the variables are $y \in \mathbb{R}^d$ and $z \in \mathbb{R}^{n-d}$. The matrix $D \in \mathbb{Q}^{d \times d}$ is diagonal and its diagonal elements satisfy $|D_{11}| \ge \cdots \ge |D_{dd}|$. Furthermore, $c \in \mathbb{Q}^d$, and $l \in \mathbb{Q}^{n-d}$. The set Λ is a lattice of rank p and dimension d, and is given via a rational lattice basis. Finally, the set $\mathcal{P} \subseteq \mathbb{R}^n$ is a polytope given via a finite system of rational linear inequalities, and it satisfies

$$\mathcal{B}(a, 1) \subset \text{proj}_y \mathcal{P} \subset \mathcal{B}(a, r_d), \tag{9}$$

where $\text{proj}_y \mathcal{P}$ denotes the orthogonal projection of $\mathcal{P}$ onto the space $\mathbb{R}^d$ of the y variables, a is a given vector in $\mathbb{Q}^d$, and r_d is an integer of size polynomial in d.

The symmetric decomposition algorithm described in Sect. 2 allows us to obtain, in strongly polynomial time, a change of basis that directly transforms (MIQP) in a separable form. In this section, our main goal is to obtain another change of basis that not only maps (MIQP) in a separable form, but also guarantees that the resulting problem is in spherical form. The additional requirements on the change of basis will result in an algorithm that is polynomial time instead of strongly polynomial. To obtain this change of basis, we rely on two key results: (i) the symmetric decomposition algorithm discussed in Sect. 2, and (ii) the existence of an algorithm based on linear programming that, for every full-dimensional polytope $\mathcal{P}$, constructs a pair of concentric ellipsoids $\mathcal{E}_1, \mathcal{E}_2$ such that $\mathcal{E}_1 \subset \mathcal{P} \subset \mathcal{E}_2$ and $\mathcal{E}_1$ is obtained by shrinking $\mathcal{E}_2$ by a factor depending only on the dimension [25].

3.1 Simultaneous diagonalization

The first result of this section does not deal directly with MIQP but is the main building block that will allow us to transform a MIQP in spherical form. This result can be interpreted as a rational version of classic theorems on simultaneous diagonalization of matrices (see Sect. 8.7 in [16]).

In order to present our result we need to introduce ellipsoids. An *ellipsoid* in $\mathbb{R}^n$ is an affine transformation of the unit ball. That is, an ellipsoid is a set of the form

$$\mathcal{E}(a, L) = \{x \in \mathbb{R}^n : \|L^{\mathsf{T}}(x - a)\| \leq 1\},$$

where $a \in \mathbb{R}^n$ and L is an $n \times n$ invertible matrix. Note that $\mathcal{B}(a, r) = \mathcal{E}(a, I_n/r)$, where I_n denotes the $n \times n$ identity matrix.

In what follows, we will often work with rational linear subspaces. In the context of polynomial time algorithms, it is not important if they are given to us via a system of linear equations or via a basis, since each description can be obtained in polynomial time from the other. Given a linear subspace $\mathcal{L}$ of $\mathbb{R}^n$ of dimension d, a *basis matrix* of $\mathcal{L}$ is an $n \times d$ matrix whose columns $b^1, \ldots, b^d$ form a basis of $\mathcal{L}$. An *$\mathcal{L}$-ellipsoid* is a set of the form

$$\mathcal{E}_{\mathcal{L}}(a, L) = \{x \in \mathcal{L} : \|L^{\mathsf{T}}(x - a)\| \leq 1\},$$

where $\mathcal{L}$ is a linear subspace of $\mathbb{R}^n$, $a \in \mathcal{L}$, and L is a basis matrix of $\mathcal{L}$. Given a linear subspace $\mathcal{L}$ of $\mathbb{R}^n$ and a set $\mathcal{S} \subseteq \mathbb{R}^n$, we denote by $\operatorname{proj}_{\mathcal{L}}(\mathcal{S})$ the orthogonal projection of $\mathcal{S}$ onto $\mathcal{L}$. We also say that a polyhedron $\{x : Wx \leq w\}$ is *rational* if W and w are rational. We are now ready to present the first result of this section.

Proposition 1 *Let H be a rational symmetric $n \times n$ matrix of rank k, let $\{x \in \mathbb{R}^n : Wx \leq w\}$ be a full-dimensional rational polytope, and let $\mathcal{M}$ be a rational linear subspace of $\mathbb{R}^n$ of dimension p. There is a polynomial time algorithm that finds a linear subspace $\mathcal{L}$ of $\mathbb{R}^n$ containing $\mathcal{M}$ and of dimension d with* $\max\{k, p\} \leq d \leq k + p$*, a $d \times d$ diagonal matrix D, and an $\mathcal{L}$-ellipsoid $\mathcal{E}_{\mathcal{L}}(a, L)$ such that*

(i) $H = LDL^{\mathsf{T}}$,
(ii) $\mathcal{E}_{\mathcal{L}}(a, L) \subset \operatorname{proj}_{\mathcal{L}}\{x : Wx \leq w\} \subset \mathcal{E}_{\mathcal{L}}(a, L/(2d^{3/2}\lceil(5d)^{d/2}\rceil^2))$.

Proof By Corollary 1 there is a strongly polynomial algorithm that computes an invertible $n \times n$ matrix L_1 and an $n \times n$ diagonal matrix D_1 such that $H = L_1 D_1 L_1^{\mathsf{T}}$. Since H has rank k and L_1 is invertible, the matrix D_1 has also rank k. Let D_2 be the matrix obtained from D_1 by deleting row i and column i for each i such that the ith diagonal element of D_1 is zero. Clearly, D_2 is an invertible $k \times k$ diagonal matrix. We also define the matrix L_2, obtained from L_1 by deleting column i for each i such that the ith diagonal element of D_1 is zero. The matrix L_2 is then an $n \times k$ matrix of rank k. Since row and column i of D_1 have all zero elements, we have $H = L_1 D_1 L_1^{\mathsf{T}} = L_2 D_2 L_2^{\mathsf{T}}$.

Let $\mathcal{L}$ be the linear subspace of $\mathbb{R}^n$ obtained as the Minkowski sum of $\mathcal{M}$ and of the linear space spanned by the k columns of L_2. Clearly, $\mathcal{L}$ contains $\mathcal{M}$, and its dimension d satisfies $\max\{k, p\} \leq d \leq k + p$. Note that $\operatorname{proj}_{\mathcal{L}}\{x : Wx \leq w\}$ is full-dimensional. It then follows form Sections 2 and 5 in [25] that there is a polynomial time algorithm which computes an $\mathcal{L}$-ellipsoid $\mathcal{E}_{\mathcal{L}}(a, C)$ such that

$$\mathcal{E}_{\mathcal{L}}(a, C) \subset \operatorname{proj}_{\mathcal{L}}\{x : Wx \leq w\} \subset \mathcal{E}_{\mathcal{L}}(a, C/(2d^{3/2})). \qquad (10)$$

Since the $n \times d$ matrix C is a basis matrix of $\mathcal{L}$ and each column of L_2 is a vector in $\mathcal{L}$, we can compute in polynomial time a $d \times k$ matrix M such that $L_2 = CM$. We obtain

$$H = L_2 D_2 L_2^\mathsf{T} = CMD_2M^\mathsf{T}C^\mathsf{T} = C\tilde{H}C^\mathsf{T},$$

where $\tilde{H} := MD_2M^\mathsf{T}$ is a $d \times d$ symmetric matrix.

By Corollary 1, applied to $\tilde{H}$, there is a strongly polynomial algorithm which computes an invertible $d \times d$ matrix $\tilde{L}$ and a $d \times d$ diagonal matrix $\tilde{D}$ such that $\tilde{H} = \tilde{L}\tilde{D}\tilde{L}^\mathsf{T}$. Furthermore, $\|\tilde{L}\|_F$ and $\|\tilde{L}^{-1}\|_F$ are upper bounded by $q_d := \lceil (5d)^{d/2} \rceil$. Note that q_d is an integer of size polynomial in d. We obtain $H = C\tilde{L}\tilde{D}\tilde{L}^\mathsf{T}C^\mathsf{T}$. By defining the $d \times d$ matrix D and the $n \times d$ matrix L in the statement as $D := \tilde{D}/q_d^2$, $L := q_d C\tilde{L}$, we obtain $H = LDL^\mathsf{T}$. Clearly, D is diagonal, thus condition (i) in the statement holds.

Note that the vector a is in $\mathcal{L}$. Moreover, since C is a basis matrix of $\mathcal{L}$ and $\tilde{L}$ is invertible, we have that also L is a basis matrix of $\mathcal{L}$. Hence $\mathcal{E}_\mathcal{L}(a, L)$ is an $\mathcal{L}$-ellipsoid. We now show that condition (ii) is satisfied. Using the fact that the Frobenius norm is submultiplicative and that $\|\tilde{L}\|_F$ and $\|\tilde{L}^{-1}\|_F$ are upper bounded by q_d, we obtain

$$\begin{aligned}
&\|C^\mathsf{T}(x-a)\| = \|\tilde{L}^{-\mathsf{T}}L^\mathsf{T}(x-a)\|/q_d \le \|\tilde{L}^{-1}\|_F \, \|L^\mathsf{T}(x-a)\|/q_d \le \|L^\mathsf{T}(x-a)\|,\\
&\|L^\mathsf{T}(x-a)\| = q_d\|\tilde{L}^\mathsf{T}C^\mathsf{T}(x-a)\| \le q_d\|\tilde{L}\|_F \, \|C^\mathsf{T}(x-a)\| \le q_d^2\|C^\mathsf{T}(x-a)\|.
\end{aligned}$$

The first chain of inequalities and (10) imply

$$\mathcal{E}_\mathcal{L}(a, L) \subseteq \mathcal{E}_\mathcal{L}(a, C) \subset \mathrm{proj}_\mathcal{L}\{x : Wx \le w\}.$$

The second chain of inequalities implies $\mathcal{E}_\mathcal{L}(a, q_d^2 C) \subseteq \mathcal{E}_\mathcal{L}(a, L)$, thus from (10),

$$\mathrm{proj}_\mathcal{L}\{x : Wx \le w\} \subset \mathcal{E}_\mathcal{L}(a, C/(2d^{3/2})) \subseteq \mathcal{E}_\mathcal{L}(a, L/(2d^{3/2}q_d^2)).$$

□

Consider now the simplest case of Proposition 1, where we set $\mathcal{M} := \mathbb{R}^n$. Then $\mathcal{L} = \mathbb{R}^n$, $d = n$, the $\mathcal{L}$-ellipsoids are just ellipsoids, and the polytope $\mathrm{proj}_\mathcal{L}\{x : Wx \le w\}$ is simply $\{x : Wx \le w\}$. In this case, Proposition 1 provides a matrix L that at the same time diagonalizes H and provides the shape of an ellipsoid that approximates the given polytope within a factor depending only on the dimension. This special case can then be interpreted as a rational version of theorems on simultaneous diagonalization of matrices. If we perform the change of basis $y := L^\mathsf{T}x$, the given matrix H is diagonalized, and the ellipsoids are just balls.

3.2 Reduction to spherical form MIQP

Next, we employ Proposition 1 to show that (MIQP) can be transformed in spherical form (S-MIQP). Throughout the paper, we denote by $e^1, e^2 \ldots, e^n$ the standard basis of $\mathbb{R}^n$.

Proposition 2 *Consider (MIQP), assume that $\{x : Wx \leq w\}$ is a full-dimensional polytope, and let k denote the rank of H. There is a polynomial time algorithm that finds a change of basis that transforms (MIQP) in spherical form (S-MIQP), where d satisfies* $\max\{k, p\} \leq d \leq k + p$*, the rank of the matrix D is k, and r_d in* (9) *is the ceiling of* $2d^{3/2}\lceil(5d)^{d/2}\rceil^2$.

Proof Consider (MIQP), assume that $\{x : Wx \leq w\}$ is a full-dimensional polytope, and let k denote the rank of H. By Proposition 1 with $\mathcal{M} := \mathbb{R}^p \times \{0\}^{n-p}$, we obtain in polynomial time a linear subspace $\mathcal{L}$ of $\mathbb{R}^n$ containing $\mathcal{M}$ and of dimension d with $\max\{k, p\} \leq d \leq k + p$, a $d \times d$ diagonal matrix D, and an $\mathcal{L}$-ellipsoid $\mathcal{E}_{\mathcal{L}}(a, L_y)$ such that $H = L_y D L_y^{\mathsf{T}}$ and

$$\mathcal{E}_{\mathcal{L}}(a, L_y) \subset \operatorname{proj}_{\mathcal{L}}\{x : Wx \leq w\} \subset \mathcal{E}_{\mathcal{L}}(a, L_y/r_d), \tag{11}$$

where we define r_d as the ceiling of $2d^{3/2}\lceil(5d)^{d/2}\rceil^2$. Since L_y is an $n \times d$ matrix of rank d, it is simple to check that the rank of D coincides with the rank of H.

We now compute an $n \times (n - d)$ basis matrix L_z of the orthogonal complement $\mathcal{L}^{\perp}$ of $\mathcal{L}$. Denote by L the $n \times n$ invertible matrix $(L_y \mid L_z)$. We perform the change of basis $x \mapsto (y, z)$, where $(y, z) \in \mathbb{R}^n$ is defined by $(y, z) := L^{\mathsf{T}}x$, i.e., $y \in \mathbb{R}^d$ is defined by $y := L_y^{\mathsf{T}}x$, and $z \in \mathbb{R}^{n-d}$ is defined by $z := L_z^{\mathsf{T}}x$.

Next, we consider the problem obtained from (MIQP) via the above change of basis, and we show that it coincides with (S-MIQP). The objective function of the new problem is

$$x^{\mathsf{T}}Hx + h^{\mathsf{T}}x = x^{\mathsf{T}}L_y D L_y^{\mathsf{T}}x + h^{\mathsf{T}}x = y^{\mathsf{T}}Dy + h^{\mathsf{T}}L^{-\mathsf{T}}(y, z),$$

which coincides with the objective function of (S-MIQP) if we define the vectors $c \in \mathbb{Q}^d$ and $l \in \mathbb{Q}^{n-d}$ by $(c, l) := L^{-1}h$. The image of the polytope $\{x : Wx \leq w\}$ is the set $\mathcal{P} := \{(y, z) : WL^{-\mathsf{T}}(y, z) \leq w\}$. Clearly, $\mathcal{P}$ is a polytope defined by a finite system of rational linear inequalities.

By definition of L_z, the linear subspace $\mathcal{L}$ can be written as $\mathcal{L} = \{x : L_z^{\mathsf{T}}x = 0\}$, thus the image of $\mathcal{L}$ under the change of basis is $\{(y, z) : z = 0\} = \mathbb{R}^d \times \{0\}^{n-d}$. Similarly, the linear subspace $\mathcal{L}^{\perp}$ can be written as $\mathcal{L}^{\perp} = \{x : L_y^{\mathsf{T}}x = 0\}$, thus the image of $\mathcal{L}^{\perp}$ is $\{0\}^d \times \mathbb{R}^{n-d}$.

Next we show that (11) implies (9). The above discussion implies that a point $\operatorname{proj}_{\mathcal{L}}(x)$ is mapped to $\operatorname{proj}_y(L^{-\mathsf{T}}(y, z)) \times \{0\}^{n-d}$. Thus, $\operatorname{proj}_{\mathcal{L}}\{x : Wx \leq w\}$ is mapped to $\operatorname{proj}_y \mathcal{P} \times \{0\}^{n-d}$. The set $\mathcal{E}_{\mathcal{L}}(a, L_y)$ is mapped to

$$\begin{aligned}\left\{(y, z) : \|y - L_y^{\mathsf{T}}a\| \leq 1\right\} \cap (\mathbb{R}^d \times \{0\}^{n-d}) &= \mathcal{E}(L_y^{\mathsf{T}}a, I_d) \times \{0\}^{n-d} \\ &= \mathcal{B}(L_y^{\mathsf{T}}a, 1) \times \{0\}^{n-d}.\end{aligned}$$

Similarly, the set $\mathcal{E}_{\mathcal{L}}(a, L_y/r_d)$ is mapped to

$$\left\{(y, z) : \|(y - L_y^{\mathsf{T}}a)/r_d\| \le 1\right\} \cap (\mathbb{R}^d \times \{0\}^{n-d}) = \mathcal{E}(L_y^{\mathsf{T}}a, I_d/r_d) \times \{0\}^{n-d}$$
$$= \mathcal{B}(L_y^{\mathsf{T}}a, r_d) \times \{0\}^{n-d}.$$

From (11), we obtain $\mathcal{B}(L_y^{\mathsf{T}}a, 1) \subset \operatorname{proj}_y \mathcal{P} \subset \mathcal{B}(L_y^{\mathsf{T}}a, r_d)$, which coincides with (9) if we redefine the vector $a \in \mathbb{Q}^d$ to be $L_y^{\mathsf{T}}a$.

We now consider the image of $\mathbb{Z}^p \times \mathbb{R}^{n-p}$. The set $\mathbb{Z}^p \times \mathbb{R}^{n-p}$ can be written as the Minkowski sum $(\mathbb{Z}^p \times \{0\}^{n-p}) + \mathcal{N} + \mathcal{L}^{\perp}$, where $\mathcal{N}$ is the orthogonal complement of $\mathcal{M}$ in $\mathcal{L}$. Since $\mathcal{M} \subseteq \mathcal{L}$ and the image of $\mathcal{L}$ is $\mathbb{R}^d \times \{0\}^{n-d}$, we have that the image of $\mathbb{Z}^p \times \{0\}^{n-p}$ is $\Lambda \times \{0\}^{n-d}$, where Λ is a lattice of rank p and dimension d. Furthermore, the image of the vectors $e^1, e^2 \dots, e^p$ forms a lattice basis $b^1, \dots, b^p$ of Λ. Since $\mathcal{N} \subseteq \mathcal{L}$, the image of $\mathcal{N}$ is $\mathcal{N}' \times \{0\}^{n-d}$, where $\mathcal{N}'$ is a linear subspace of $\mathbb{R}^d$ of dimension $d - p$. Since $\mathcal{M}$ and $\mathcal{N}$ are orthogonal, we have that $\Lambda + \mathcal{N}'$ has dimension d. Finally, we know that the image of $\mathcal{L}^{\perp}$ is $\{0\}^d \times \mathbb{R}^{n-d}$. We conclude that the image of $\mathbb{Z}^p \times \mathbb{R}^{n-p}$ is $(\Lambda + \mathcal{N}') \times \mathbb{R}^{n-d}$. Let Λ' be the orthogonal projection of Λ onto $\mathcal{N}'^{\perp}$. Then Λ' is a lattice of rank p and dimension d, and the image of $\mathbb{Z}^p \times \mathbb{R}^{n-p}$ is $(\Lambda' + \operatorname{span}(\Lambda')^{\perp}) \times \mathbb{R}^{n-d}$ as desired. A basis of Λ' can be obtained by taking the orthogonal projection of $b^1, \dots, b^p$ onto $\mathcal{N}'^{\perp}$.

By eventually reordering the components of the vector y, and accordingly the data of the problem, we obtain that the diagonal elements of the matrix D satisfy $|D_{11}| \ge \cdots \ge |D_{dd}|$. □

Next, we briefly discuss how Proposition 2 simplifies in the pure integer setting and in the pure continuous setting. In the pure integer setting we have $p = n$ in (MIQP), and Proposition 2 implies $d = n$. Therefore, in (S-MIQP) we have no z variables and the constraint $y \in \Lambda + \operatorname{span}(\Lambda)^{\perp}$ is replaced by $y \in \Lambda$ since the set Λ is a full rank lattice of dimension n. Furthermore, in (9), the set $\operatorname{proj}_y \mathcal{P}$ is replaced by $\mathcal{P}$. In the pure continuous setting we have $p = 0$ in (MIQP), and Proposition 2 implies $d = k$. Therefore, in (S-MIQP) the constraint $y \in \Lambda + \operatorname{span}(\Lambda)^{\perp}$ is replaced by $y \in \mathbb{R}^d$ since the set Λ is a lattice of rank zero.

We remark that a change of basis similar to the one given by Proposition 2 can be obtained through the use of eigenvalue methods like the Schur decomposition [16], instead of our symmetric decomposition algorithm. These techniques have been used by Vavasis to obtain a related change of basis for QP (see page 282 in [30]). Unfortunately, these methods do not yield polynomial time algorithms since symmetric integer matrices can have irrational eigenvalues.

4 Aligned vectors

In this section, we introduce the notion of aligned vectors. Given an instance of problem (S-MIQP), two vectors $y^+, y^- \in \mathbb{R}^d$ are said to be *aligned* if $y^+, y^- \in \mathcal{B}(a, 1) \cap (2\Lambda + \operatorname{span}(\Lambda)^{\perp})$, and $y_1^+ - y_1^- \ge 1$, $\sum_{i=2}^d (y_i^+ - y_i^-)^2 \le 1/4$. The end goal of this section is to show that, if there exist two aligned vectors, then, for every $\epsilon \in (0, 1]$, it is possible to find an ϵ-approximate solution to (S-MIQP) by solving a number of MILPs.

We begin by showing, in Lemma 4, how aligned vectors allow us to obtain a lower bound on the gap between maximum and minimum of a separable quadratic function evaluated on the two vectors and their midpoint. In the proof of Lemma 4 we use the following simple lemma. The proof is that of Lemma 3 in [30], even though our statement is slightly stronger.

Lemma 3 *Let $q(\lambda) = a\lambda^2 + b\lambda + c$ be a univariate quadratic function and let $u, \ell \in \mathbb{R}$. Let $\underline{q}$ and $\overline{q}$ be the minimum and maximum values attained by q on the three points $u, \ell, (u+\ell)/2$. Then $\overline{q} - \underline{q} \geq |a|(u-\ell)^2/4$.*

Lemma 4 *Let $f : \mathbb{R}^d \times \mathbb{R}^{n-d} \to \mathbb{R}$ be a quadratic function of the form $f(y,z) = y^T D y + c^T y + l^T z$, where D is diagonal and D_{11} is the element of D with the largest absolute value. Let $(y^+, z^+), (y^-, z^-) \in \mathbb{R}^d \times \mathbb{R}^{n-d}$ such that $y_1^+ - y_1^- \geq 1$ and $\sum_{i=2}^d (y_i^+ - y_i^-)^2 \leq 1/4$. Let $\underline{f}$ and $\overline{f}$ be the minimum and maximum values attained by f on the three vectors $(y^+, z^+), (y^-, z^-), (y^+, z^+)/2 + (y^-, z^-)/2$. Then $\overline{f} - \underline{f} \geq \frac{3}{16}|D_{11}|$.*

Proof By eventually replacing f with $-f$, we can assume without loss of generality that $D_{11} \geq 0$. Let $q : \mathbb{R} \to \mathbb{R}$ be defined by

$$q(\lambda) := f\left((y^-, z^-) + \lambda\left((y^+, z^+) - (y^-, z^-)\right)\right).$$

Using the separability of f, we obtain

$$q(\lambda) = \sum_{i=1}^d D_{ii}\left(y_i^- + \lambda(y_i^+ - y_i^-)\right)^2 + O(\lambda) = \lambda^2 \cdot \sum_{i=1}^d D_{ii}(y_i^+ - y_i^-)^2 + O(\lambda).$$

To conclude the proof we just need to show that

$$\left|\sum_{i=1}^d D_{ii}(y_i^+ - y_i^-)^2\right| \geq \frac{3}{4} D_{11}. \tag{12}$$

In fact, by noting that $q(0) = f(y^-, z^-)$, $q(1) = f(y^+, z^+)$, and $q(1/2) = f\left((y^+, z^+)/2 + (y^-, z^-)/2\right)$, we can apply Lemma 3 to q and the points $0, 1 \in \mathbb{R}$ and obtain

$$\overline{f} - \underline{f} = \overline{q} - \underline{q} \geq \frac{1}{4}\left|\sum_{i=1}^d D_{ii}(y_i^+ - y_i^-)^2\right| \geq \frac{3}{16} D_{11}.$$

To prove inequality (12), we bound its left hand side as follows:

$$\left|\sum_{i=1}^{d} D_{ii}(y_i^+ - y_i^-)^2\right| \geq \sum_{i=1}^{d} D_{ii}(y_i^+ - y_i^-)^2 =$$

$$= \sum_{i:D_{ii}\geq 0} D_{ii}(y_i^+ - y_i^-)^2 - \sum_{i:D_{ii}<0} -D_{ii}(y_i^+ - y_i^-)^2.$$

We can now separately bound the two nonnegative sums using the assumption on D_{11}, and the conditions $y_1^+ - y_1^- \geq 1$ and $\sum_{i=2}^{d}(y_i^+ - y_i^-)^2 \leq 1/4$.

$$\sum_{i:D_{ii}\geq 0} D_{ii}(y_i^+ - y_i^-)^2 \geq D_{11}(y_1^+ - y_1^-)^2 \geq D_{11},$$

$$\sum_{i:D_{ii}<0} -D_{ii}(y_i^+ - y_i^-)^2 \leq D_{11} \sum_{i:D_{ii}<0} (y_i^+ - y_i^-)^2 \leq D_{11}/4.$$

Hence inequality (12) holds. □

We are now ready to discuss our approximation algorithm for spherical form MIQPs for which there exist two aligned vectors. This algorithm is based on the classic technique of mesh partition and linear underestimators. This natural approach consists in replacing the nonlinear objective function with a piecewise linear approximation, an idea known in the field of optimization since at least the 1950s. Mesh partition and linear underestimators have proven to be a very successful technique to obtain approximation algorithms for several special classes of MIQP [7–9, 30, 31]. In this section, for the first time we employ mesh partition and linear underestimators to MIQPs that, at the same time, have integer variables and an indefinite quadratic objective function. The generality of this setting poses a number of additional challenges, and the results presented in the paper so far provide the key to successfully apply these techniques. In the proof, we will use the following standard lemma.

Lemma 5 *Let $q(\lambda) = a\lambda^2 + b\lambda + c$ be a univariate quadratic function and let $u, \ell \in \mathbb{R}$. Let $q'(\lambda)$ be the affine univariate function that attains the same values as q at ℓ, u. Then $|q'(\lambda) - q(\lambda)| \leq |a|(u-\ell)^2/4$ for every $\lambda \in [\ell, u]$.*

Proposition 3 *Consider (S-MIQP), assume that there exist two aligned vectors, and let k be the rank of the matrix D. For every $\epsilon \in (0, 1]$, there is an algorithm that finds an ϵ-approximate solution, if it exists, by solving at most $\left\lceil 4r_d\sqrt{k/(3\epsilon)}\right\rceil^k$ MILPs of the same size as (S-MIQP) and with p integer variables.*

Proof We start by describing the approximation algorithm. We define φ^k boxes in $\mathbb{R}^k$, where $\varphi := \left\lceil 4r_d\sqrt{k/(3\epsilon)}\right\rceil$:

$$\mathcal{C}_{j_1,\dots,j_k} := \prod_{i=1}^{k}\left(\{a_i - r_d\} + \frac{2r_d}{\varphi}[j_i - 1, j_i]\right) \quad \forall j_1,\dots,j_k \in \{1,\dots,\varphi\}. \tag{13}$$

Note that the union of these φ^k boxes is the polytope

$$\{(y_1, \dots, y_k) \in \mathbb{R}^k : a_i - r_d \le y_i \le a_i + r_d \ \forall i = 1, \dots, k\},$$

which contains the projection of $\mathcal{P}$ onto the space defined by the first k coordinates of y, since $\operatorname{proj}_y \mathcal{P} \subset \mathcal{B}(a, r_d)$ from (9).

For each box $\mathcal{C} = \prod_{i=1}^{k} [\ell_i, u_i]$ among those defined in (13), we construct the affine functions $g_i : \mathbb{R} \to \mathbb{R}$ that attain the same values as $D_{ii} y_i^2$ at ℓ_i, u_i, for $i = 1, \dots, k$:

$$g_i(y_i) := D_{ii}(\ell_i + u_i) y_i - D_{ii} \ell_i u_i \qquad \forall i = 1, \dots, k.$$

We define $\gamma := |D_{11}|$. Then we define the affine function $g : \mathbb{R}^k \to \mathbb{R}$ given by

$$g(y_1, \dots, y_k) := \sum_{i=1}^{k} g_i(y_i) - \frac{\gamma r_d^2}{\varphi^2} |\{i \in \{1, \dots, k\} : D_{ii} > 0\}|. \tag{14}$$

We solve the MILP obtained from (S-MIQP) by substituting $y^\mathsf{T} D y$ with $g(y_1, \dots, y_k)$ and adding the constraint $(y_1, \dots, y_k) \in \mathcal{C}$:

$$\begin{aligned} \min \quad & g(y_1, \dots, y_k) + c^\mathsf{T} y + l^\mathsf{T} z \\ \text{s. t.} \quad & (y, z) \in \mathcal{P} \\ & (y_1, \dots, y_k) \in \mathcal{C} \\ & y \in \Lambda + \operatorname{span}(\Lambda)^\perp, \ z \in \mathbb{R}^{n-d}. \end{aligned} \tag{15}$$

To see that (15) is indeed a MILP, one just needs to perform a change of basis that maps Λ to $\mathbb{Z}^p \times \{0\}^{d-p}$ and $\operatorname{span}(\Lambda)^\perp$ to $\{0\}^p \times \mathbb{R}^{d-p}$.

The approximation algorithm returns the best solution $(y^\diamond, z^\diamond)$ among all the (at most) φ^k optimal solutions just obtained of the MILPs (15). If all the MILPs (15) are infeasible, the algorithm returns that (S-MIQP) is infeasible. This concludes the description of the algorithm.

Next, we show that $(y^\diamond, z^\diamond)$ is an ϵ-approximate solution to (S-MIQP). To simplify the notation, in this proof we denote the objective function of (S-MIQP) by

$$f(y, z) := y^\mathsf{T} D y + c^\mathsf{T} y + l^\mathsf{T} z = \sum_{i=1}^{k} D_{ii} y_i^2 + c^\mathsf{T} y + l^\mathsf{T} z.$$

In order to show that $(y^\diamond, z^\diamond)$ is an ϵ-approximate solution, we derive two bounds: (i) an upper bound on $f(y^\diamond, z^\diamond) - f(y^*, z^*)$, where (y^*, z^*) is an optimal solution to (S-MIQP), and (ii) a lower bound on $f_{\max} - f(y^*, z^*)$, where $f_{\max}$ is the maximum value of $f(y, z)$ on the feasible region of (S-MIQP). Note that both bounds will depend linearly on γ. This dependence is what allows us to solve a number of MILPs that is independent on γ.

An upper bound on $f(y^\diamond, z^\diamond) - f(y^*, z^*)$. Let $\mathcal{C} \subset \mathbb{R}^k$ be a box constructed in (13), say $\mathcal{C} = \prod_{i=1}^k [\ell_i, u_i]$. For each $i = 1, \dots, k$, we apply Lemma 5 to each univariate quadratic function $D_{ii} y_i^2$ and points ℓ_i, u_i. Since $u_i - \ell_i = 2r_d/\varphi$ and $|D_{ii}| \le \gamma$ for $i = 1, \dots, k$, we obtain that, for every $(y_1, \dots, y_k) \in \mathcal{C}$,

$$\begin{aligned} g_i(y_i) - \gamma r_d^2/\varphi^2 \le D_{ii} y_i^2 &\le g_i(y_i) && \text{if } D_{ii} > 0 \\ g_i(y_i) \le D_{ii} y_i^2 &\le g_i(y_i) + \gamma r_d^2/\varphi^2 && \text{if } D_{ii} < 0. \end{aligned}$$

We sum up all these inequalities for $i = 1, \dots, k$ and obtain that for every $(y_1, \dots, y_k) \in \mathcal{C}$,

$$g(y_1, \dots, y_k) \le \sum_{i=1}^k D_{ii} y_i^2 \le g(y_1, \dots, y_k) + \gamma k r_d^2/\varphi^2. \tag{16}$$

Let $\mathcal{C}^\diamond \subset \mathbb{R}^k$ be the box constructed in (13) that yields the solution $(y^\diamond, z^\diamond)$ and let $g^\diamond$ be the corresponding affine function defined in (14). Let $\mathcal{C}^* \subset \mathbb{R}^k$ be a box constructed in (13) such that $(y^*, z^*) \in \mathcal{C}^*$ and let g^* be the corresponding affine function. We have

$$\begin{aligned} f(y^\diamond, z^\diamond) &\le g^\diamond(y_1^\diamond, \dots, y_k^\diamond) + c^\mathsf{T} y^\diamond + l^\mathsf{T} z^\diamond + \gamma k r_d^2/\varphi^2 \\ &\le g^*(y_1^*, \dots, y_k^*) + c^\mathsf{T} y^* + l^\mathsf{T} z^* + \gamma k r_d^2/\varphi^2 \\ &\le f(y^*, z^*) + \gamma k r_d^2/\varphi^2. \end{aligned} \tag{17}$$

The first inequality follows by applying the right inequality in (16) to $\mathcal{C}^\diamond$ and $y^\diamond$. The second inequality holds by definition of $(y^\diamond, z^\diamond)$. The third inequality follows by applying the left inequality in (16) to $\mathcal{C}^*$ and y^*.

A lower bound on $f_{\max} - f(y^*, z^*)$. By assumption, there exist two aligned vectors y^+, y^- for (S-MIQP). From (9) we have $\mathcal{B}(a, 1) \subset \operatorname{proj}_y \mathcal{P}$, thus there exist $z^+, z^- \in \mathbb{R}^{n-d}$ such that the vectors $(y^+, z^+), (y^-, z^-) \in \mathbb{R}^d \times \mathbb{R}^{n-d}$ are in $\mathcal{P}$. We define the midpoint of the segment joining (y^+, z^+) and (y^-, z^-) as $(y^\circ, z^\circ) := (y^+, z^+)/2 + (y^-, z^-)/2$. By convexity, the vector (y°, z°) is in $\mathcal{P}$. Moreover, as both vectors $y^+/2$, $y^-/2$ are in $\Lambda + \operatorname{span}(\Lambda)^\perp$, so is their sum y°. Let $\underline{f}$ and $\overline{f}$ be the minimum and maximum values attained by f on the three vectors (y^+, z^+), (y^-, z^-), (y°, z°). Then, by Lemma 4, $\overline{f} - \underline{f} \ge \frac{3}{16}|D_{11}| = \frac{3}{16}\gamma$. Since all three vectors are feasible to (S-MIQP), we conclude that

$$f_{\max} - f(y^*, z^*) \ge \frac{3}{16}\gamma. \tag{18}$$

We are now ready to show that $(y^\diamond, z^\diamond)$ is an ϵ-approximate solution to (S-MIQP). We have

$$\frac{f(y^\diamond, z^\diamond) - f(y^*, z^*)}{f_{\max} - f(y^*, z^*)} \le \frac{\cancel{\gamma} k r_d^2}{\varphi^2} \cdot \frac{16}{3\cancel{\gamma}} = \frac{16}{3}\frac{k r_d^2}{\varphi^2} \le \epsilon.$$

In the first inequality we used (17) and (18), and the last inequality holds by the definition of φ given at the beginning of the proof. □

In particular, note that the number of MILPs solved in Proposition 3 is polynomial in $1/\epsilon$ if k and d are fixed. Due to Proposition 2, this is indeed the case if both k and p are fixed in the original (MIQP).

5 Flatness and decomposition of spherical form MIQP

In Sect. 4 we saw that, if a spherical form MIQP has two aligned vectors, then we can find an ϵ-approximate solution. But what if there are no aligned vectors? In this section, we show that in this case we can decompose the problem in a number of MIQPs with fewer integer variables. This result will play a crucial role in our approximation algorithm for MIQP. Before stating our theorem, we recall the concepts of width and of reduced basis.

Let $\mathcal{S} \subseteq \mathbb{R}^d$ be a bounded closed convex set. Given a vector $v \in \mathbb{R}^d$, we define the *width of* $\mathcal{S}$ *along* v to be

$$\text{width}_v(\mathcal{S}) := \max\{v^\mathsf{T} y : y \in \mathcal{S}\} - \min\{v^\mathsf{T} y : y \in \mathcal{S}\}.$$

Let Λ be a lattice of rank p and dimension d, and let $b^1, \dots, b^p \in \mathbb{R}^d$ be a lattice basis of Λ. Consider now a vector $v \in \mathbb{R}^d$ that satisfies $v^\mathsf{T} b^i \in \mathbb{Z}$ for every $i = 1, \dots, p$. Then $v^\mathsf{T} y$ is an integer for every $y \in \Lambda$ since y can be written as an integer linear combination of the b^i. It follows that $\text{width}_v(\mathcal{S})$ is an upper bound on the number of hyperplanes orthogonal to v that contain points in $\mathcal{S} \cap \Lambda$.

Next, we recall the notion of reduced basis. Let Λ be a lattice of rank p and dimension d, and let $b^1, \dots, b^p \in \mathbb{R}^d$ be a lattice basis of Λ. The $d \times p$ matrix B formed by taking the columns to be the basis vectors b^i is called a *basis matrix* of Λ. The *determinant* of Λ is the volume of the fundamental parallelepiped of any basis for Λ, that is, $\det(\Lambda) := \sqrt{\det(B^\mathsf{T} B)}$.

Lovász introduced the notion of a reduced basis, using a Gram-Schmidt orthogonal basis as a reference. The *Gram-Schmidt procedure* is as follows. Define $g^1 := b^1$ and, recursively, for $i = 2, \dots, p$, define $g^i \in \mathbb{R}^d$ as the projection of b^i onto the orthogonal complement of the linear space spanned by $b^1, \dots, b^{i-1}$. Formally, for $i = 2, \dots, p$, g^i is defined by

$$g^i := b^i - \sum_{j=1}^{i-1} \mu_{ij} g^j, \qquad \text{where } \mu_{ij} := \frac{(b^i)^\mathsf{T} g^j}{\|g^j\|^2} \qquad \forall j = 1, \dots, i-1. \tag{19}$$

By construction, the Gram-Schmidt basis $g^1, \dots, g^p$ is an orthogonal basis of $\text{span}(\Lambda)$ with the property that, for $i = 1, \dots, p$, the linear spaces spanned by $b^1, \dots, b^i$ and by $g^1, \dots, g^i$ coincide. Moreover, we have $\|b^i\| \geq \|g^i\|$ for $i = 1, \dots, p$, and $\|g^1\| \cdots \|g^p\| = \det(\Lambda)$.

A basis $r^1, \dots, r^p$ of the lattice Λ is said to be *reduced* if it satisfies the following two conditions

$$\begin{aligned} |\mu_{ij}| &\le \frac{1}{2} && \text{for } 1 \le j < i \le p \\ \|g^i + \mu_{i,i-1} g^{i-1}\|^2 &\ge \frac{3}{4}\|g^{i-1}\|^2 && \text{for } 2 \le i \le p, \end{aligned}$$

where $g^1, \dots, g^p$ is the output of the Gram-Schmidt procedure when applied to $r^1, \dots, r^p$. Lovász' celebrated *basis reduction algorithm* yields a reduced basis, and it runs in polynomial time in the size of the original basis. If a basis $r^1, \dots, r^p$ of Λ is reduced, then it is "nearly orthogonal", in the sense that it satisfies

$$\|r^1\| \cdots \|r^p\| \le 2^{p(p-1)/4} \det(\Lambda). \tag{20}$$

See for example [2] for more details on lattices and reduced basis, or [14] for an exposition that does not consider only full rank lattices.

In order to show our decomposition result for spherical form MIQP, we first prove the following Lenstra-type proposition.

Proposition 4 *Let $a \in \mathbb{Q}^d$, $\delta \in \mathbb{Q}$ with $\delta \ge 0$, and let Λ be a lattice of rank p and dimension d with basis matrix $B \in \mathbb{Q}^{d \times p}$. There is a polynomial time algorithm which either finds a vector $\bar{y} \in \mathcal{B}(a, \delta) \cap (\Lambda + \operatorname{span}(\Lambda)^\perp)$, or finds a vector $v \in \operatorname{span}(\Lambda)$ with $v^T B$ integer such that $\operatorname{width}_v(\mathcal{B}(a, \delta)) \le p2^{p(p-1)/4}$.*

Proof If $p = 0$, then the algorithm simply returns $\bar{y} = a$, thus we now assume $p \ge 1$. The basis reduction algorithm gives in polynomial time a reduced basis $r^1, \dots, r^p \in \mathbb{Q}^d$ of the lattice Λ. Let $\hat{r}^1, \dots \hat{r}^p \in \mathbb{Q}^d$ be obtained by reordering $r^1, \dots r^p$ so that the vector in the last position has maximum norm, and denote by $\hat{R} \in \mathbb{Q}^{d \times p}$ the corresponding basis matrix. Since B and $\hat{R}$ are basis matrices of the same lattice Λ, it is well known that we can find in polynomial time a $p \times p$ unimodular matrix U such that $B = \hat{R}U$.

Let $a_\Lambda := \operatorname{proj}_{\operatorname{span}(\Lambda)} a \in \mathbb{Q}^d$, let $\lambda \in \mathbb{Q}^p$ be such that $\hat{R}\lambda = a_\Lambda$, and define $y_\Lambda := \hat{R}\lfloor\lambda\rceil \in \mathbb{Q}^d$, where $\lfloor\lambda\rceil = (\lfloor\lambda_1\rceil, \dots, \lfloor\lambda_p\rceil)$ and $\lfloor\lambda_i\rceil$ denotes a nearest integer to λ_i. Clearly, $y_\Lambda \in \Lambda$. Consider first the case $y_\Lambda \in \operatorname{proj}_{\operatorname{span}(\Lambda)}(\mathcal{B}(a, \delta))$. This implies that the vector $\bar{y} := (a + \operatorname{span}(\Lambda)) \cap (y_\Lambda + \operatorname{span}(\Lambda)^\perp)) \in \mathbb{Q}^d$ is in $\mathcal{B}(a, \delta)$. Since $y_\Lambda \in \Lambda$, we have that $\bar{y} \in \Lambda + \operatorname{span}(\Lambda)^\perp$. Therefore, in this case we are done. Hence, in the remainder of the proof we consider the case $y_\Lambda \notin \operatorname{proj}_{\operatorname{span}(\Lambda)}(\mathcal{B}(a, \delta))$.

Since B is a $d \times p$ matrix of rank p, the matrix $B^\mathsf{T} B$ is an invertible $p \times p$ symmetric matrix, thus we can define the $p \times d$ matrix $B^\dagger := (B^\mathsf{T} B)^{-1} B^\mathsf{T}$. The matrix $B^\dagger$ is a left inverse of B, i.e., $B^\dagger B$ is the identity matrix I_p. Let $u \in \mathbb{Z}^{1 \times p}$ be the last row of U, and define the vector $v := (uB^\dagger)^\mathsf{T} \in \mathbb{Q}^d$. We have that $v \in \operatorname{span}(\Lambda)$, since for every vector $t \in (\operatorname{span}(\Lambda))^\perp$ we have $v^\mathsf{T} t = uB^\dagger t = u(B^\mathsf{T} B)^{-1} B^\mathsf{T} t = 0$, since each column of B lies in $\operatorname{span}(\Lambda)$. Moreover, the vector $v^\mathsf{T} B$ is integer since $v^\mathsf{T} B = uB^\dagger B = uI_p = u$. Hence, to complete the proof, we only need to show $\operatorname{width}_v(\mathcal{B}(a, \delta)) \le p2^{p(p-1)/4}$.

The assumption $y_\Lambda \notin \operatorname{proj}_{\operatorname{span}(\Lambda)}(\mathcal{B}(a, \delta))$ is equivalent to $\|y_\Lambda - a_\Lambda\| > \delta$. Since $y_\Lambda = \hat{R}\lfloor\lambda\rceil$ and $a_\Lambda = \hat{R}\lambda$, we have

$$\begin{aligned}\|y_\Lambda - a_\Lambda\| = \|\hat{R}(\lfloor\lambda\rceil - \lambda)\| &= \left\| \sum_{i=1}^{p} (\lfloor\lambda_i\rceil - \lambda_i)\hat{r}^i \right\| \\ &\le \sum_{i=1}^{p} |\lfloor\lambda_i\rceil - \lambda_i| \; \|\hat{r}^i\| \le p\|\hat{r}^p\|/2.\end{aligned}$$

We obtain that $\|\hat{r}^p\| > 2\delta/p$. Consider the Gram-Schmidt orthogonal basis $\hat{g}^1, \dots, \hat{g}^p \in \mathbb{Q}^d$ obtained from $\hat{r}^1, \dots, \hat{r}^p$. Using (20) we have

$$\|\hat{r}^1\| \cdots \|\hat{r}^p\| = \|r^1\| \cdots \|r^p\| \le 2^{p(p-1)/4} \det(\Lambda) = 2^{p(p-1)/4} \|\hat{g}^1\| \cdots \|\hat{g}^p\|.$$

Moreover, as $\|\hat{r}^i\| \ge \|\hat{g}^i\|$ for $i = 1, \dots, p-1$, it follows that $\|\hat{r}^p\| \le 2^{p(p-1)/4}\|\hat{g}^p\|$. Since $\|\hat{r}^p\| > 2\delta/p$, we obtain

$$\|\hat{g}^p\| > \frac{2\delta}{p2^{p(p-1)/4}}. \tag{21}$$

We define the $p \times d$ matrix $\hat{R}^\dagger := (\hat{R}^\mathsf{T}\hat{R})^{-1}\hat{R}^\mathsf{T}$, which is a left inverse of $\hat{R}$. Using $B = \hat{R}U$, we obtain the relation

$$\begin{aligned}B^\dagger &= (B^\mathsf{T}B)^{-1}B^\mathsf{T} = (U^\mathsf{T}\hat{R}^\mathsf{T}\hat{R}U)^{-1}U^\mathsf{T}\hat{R}^\mathsf{T} \\ &= U^{-1}(\hat{R}^\mathsf{T}\hat{R})^{-1}U^{-\mathsf{T}}U^\mathsf{T}\hat{R}^\mathsf{T} = U^{-1}(\hat{R}^\mathsf{T}\hat{R})^{-1}\hat{R}^\mathsf{T} = U^{-1}\hat{R}^\dagger.\end{aligned}$$

It is simple to check that $\operatorname{width}_v(\mathcal{B}(a, \delta)) = 2\delta\|v\|$. If we denote by $\hat{r} \in \mathbb{Q}^{1\times d}$ the last row of $\hat{R}^\dagger$, we have

$$\operatorname{width}_v(\mathcal{B}(a, \delta)) = 2\delta\|v\| = 2\delta\|(uB^\dagger)^\mathsf{T}\| = 2\delta\|(uU^{-1}\hat{R}^\dagger)^\mathsf{T}\| = 2\delta\|\hat{r}^\mathsf{T}\|, \tag{22}$$

where the last equality holds since u is the last row of U.

We now show that $\hat{r}^\mathsf{T} = \hat{g}^p/\|\hat{g}^p\|^2$. First, note that both $\hat{r}^\mathsf{T}$ and $\hat{g}^p$ live in $\operatorname{span}(\Lambda)$. For $\hat{g}^p$ this follows from the fact that $\hat{g}^1, \dots, \hat{g}^p$ is a basis of $\operatorname{span}(\Lambda)$. For $\hat{r}^\mathsf{T}$, it can be seen because this vector is orthogonal to each vector $t \in (\operatorname{span}(\Lambda))^\perp$ as $\hat{r}$ is the last row of $\hat{R}^\dagger$ and we have $\hat{R}^\dagger t = (\hat{R}^\mathsf{T}\hat{R})^{-1}\hat{R}^\mathsf{T}t = 0$, since each column of $\hat{R}$ lies in $\operatorname{span}(\Lambda)$. Since $\hat{g}^p$ is orthogonal to $\hat{g}^1, \dots, \hat{g}^{p-1}$, it follows from (19) that $(\hat{g}^p)^\mathsf{T}\hat{r}^i = 0$ for $i = 1, \dots, p-1$ and $(\hat{g}^p)^\mathsf{T}\hat{r}^p = \|\hat{g}^p\|^2$. Since $\hat{r}$ is the last row of $\hat{R}^\dagger$, we have $\hat{r}\hat{r}^i = 0$ for $i = 1, \dots, p-1$ and $\hat{r}\hat{r}^p = 1$. This concludes the proof that $\hat{r}^\mathsf{T} = \hat{g}^p/\|\hat{g}^p\|^2$.

Thus, by (22) and (21),

$$\operatorname{width}_v(\mathcal{B}(a, \delta)) = 2\delta\|\hat{r}^\mathsf{T}\| = \frac{2\delta}{\|\hat{g}^p\|} \le p2^{p(p-1)/4}.$$

□

We are now ready to give our decomposition result.

Proposition 5 *There is a polynomial time algorithm which either finds two aligned vectors for (S-MIQP), or finds a vector* $v \in \operatorname{span}(\Lambda)$ *with* $v^T B$ *integer such that* $\operatorname{width}_v(\mathcal{P}) \le r_d s_p$*, where* $s_p := 14p2^{p(p-1)/4}$.

Proof Let $a^+ := a + \frac{3}{4}e^1 \in \mathbb{Q}^d$, where e^1 denotes the first vector of the standard basis of $\mathbb{R}^d$. It is simple to verify that

$$\mathcal{B}(a^+, 1/4) \subseteq \mathcal{B}(a, 1) \subseteq \mathcal{B}(a^+, 7/4). \tag{23}$$

Denote by $B \in \mathbb{Q}^{d\times p}$ the given basis matrix of the lattice Λ. We apply Proposition 4 to $\mathcal{B}(a^+, 1/4)$ and the lattice 2Λ with basis matrix $2B$. Consider first the case where Proposition 4 finds a vector $v \in \operatorname{span}(\Lambda)$ with $v^\mathsf{T}(2B)$ integer such that $\operatorname{width}_v(\mathcal{B}(a^+, 1/4)) \le p2^{p(p-1)/4}$. We then set $v' := 2v$ and note that $v' \in \operatorname{span}(\Lambda)$ with $v'^\mathsf{T} B$ integer. Furthermore, it follows from (23) that

$$\begin{aligned}\operatorname{width}_{v'}(\mathcal{B}(a, 1)) &= 2\operatorname{width}_v(\mathcal{B}(a, 1)) \le 2\operatorname{width}_v(\mathcal{B}(a^+, 7/4))\\ &= 14\operatorname{width}_v(\mathcal{B}(a^+, 1/4)) \le 14p2^{p(p-1)/4} = s_p.\end{aligned}$$

Using (9) we obtain

$$\begin{aligned}\operatorname{width}_{v'}(\mathcal{P}) &= \operatorname{width}_{v'}(\operatorname{proj}_y \mathcal{P})) \le \operatorname{width}_{v'}(\mathcal{B}(a, r_d))\\ &\le r_d \operatorname{width}_{v'}(\mathcal{B}(a, 1)) \le r_d s_p.\end{aligned}$$

Hence the statement of the proposition holds. Therefore, we now assume that Proposition 4 finds a vector $y^+ \in \mathcal{B}(a^+, 1/4) \cap (2\Lambda + \operatorname{span}(\Lambda)^\perp)$. Clearly, (23) implies that $y^+ \in \mathcal{B}(a, 1)$.

Next, we define $a^- := a - \frac{3}{4}e^1 \in \mathbb{Q}^d$, and we apply Proposition 4 to $\mathcal{B}(a^-, 1/4)$ and the lattice 2Λ with basis matrix $2B$. Symmetrically, we can assume that Proposition 4 finds a vector $y^- \in 2\Lambda + \operatorname{span}(\Lambda)^\perp$ that is in $\mathcal{B}(a^-, 1/4)$ and therefore in $\mathcal{B}(a, 1)$.

To conclude the proof, we only need to show that the vectors y^+, y^- are aligned for (S-MIQP). Since $y^+ \in \mathcal{B}(a^+, 1/4)$ and $y^- \in \mathcal{B}(a^-, 1/4)$, we obtain $y_1^+ - y_1^- \ge (a_1 + 1/2) - (a_1 - 1/2) = 1$. For a vector $y \in \mathbb{R}^d$ we denote by y_{-1} the vector in $\mathbb{R}^{d-1}$ obtained by deleting the first component from y. Using the triangle inequality and the fact that $a^+_{-1} = a^-_{-1} = a_{-1}$, we obtain

$$\begin{aligned}\sum_{i=2}^{d}(y_i^+ - y_i^-)^2 &= \|y^+_{-1} - y^-_{-1}\|^2 \le (\|y^+_{-1} - a_{-1}\| + \|y^-_{-1} - a_{-1}\|)^2\\ &= (\|y^+_{-1} - a^+_{-1}\| + \|y^-_{-1} - a^-_{-1}\|)^2 \le (\|y^+ - a^+\| + \|y^- - a^-\|)^2\\ &\le (1/4 + 1/4)^2 = 1/4.\end{aligned}$$

Hence y^+, y^- are aligned for (S-MIQP). □

6 Approximation algorithm

In this section, we present our approximation algorithm for (MIQP) and we prove Theorem 1. First, we present two lemmas that allow us to reduce the number of variables in MIQPs with polyhedra that are not full-dimensional. The arguments are direct extensions of those for pure integer MILPs (see, e.g., [2]). Proofs are given for completeness.

Lemma 6 *Let $a \in \mathbb{Q}^n \setminus \{0\}$, $\beta \in \mathbb{Q}$, $p \in \{0, \dots, n\}$. There is a polynomial time algorithm that determines whether the set $\mathcal{S} := \{x \in \mathbb{Z}^p \times \mathbb{R}^{n-p} : a^T x = \beta\}$ is empty or not. If $\mathcal{S} \neq \emptyset$, the algorithm finds a vector $\bar{x} \in \mathbb{Q}^n$ and a matrix $M \in \mathbb{Q}^{n\times(n-1)}$ such that*

$$\begin{aligned} &\mathcal{S} = \{\bar{x} + My : y \in \mathbb{Z}^p \times \mathbb{R}^{n-p-1}\} && \text{if } a_i \neq 0 \text{ for some } i \in \{p+1, \dots, n\} \\ &\mathcal{S} = \{\bar{x} + My : y \in \mathbb{Z}^{p-1} \times \mathbb{R}^{n-p}\} && \text{if } a_i = 0 \text{ for all } i \in \{p+1, \dots, n\}. \end{aligned}$$

Proof First, consider the case $a_i \neq 0$ for some $i \in \{p+1, \dots, n\}$. We can then rewrite $a^\mathsf{T} x = \beta$ in the form $x_i = (\beta - \sum_{j\in\{1,\dots,n\}\setminus\{i\}} a_j x_j)/a_i$. Since x_i is a continuous variable, we obtain that $\mathcal{S}$ is nonempty. We define the vector $\bar{x} \in \mathbb{Q}^n$ with entry $\bar{x}_i := \beta/a_i$ and all other entries zero. We also define the matrix $M \in \mathbb{Q}^{n\times n-1}$ obtained from the $n \times n$ identity matrix by replacing the ith row with the horizontal vector $-a^\mathsf{T}/a_i$ and deleting column i. With these definitions of $\bar{x}$ and M, we obtain

$$\mathcal{S} = \{\bar{x} + My : y \in \mathbb{Z}^p \times \mathbb{R}^{n-p-1}\}.$$

Next, consider the case $a_i = 0$ for all $i \in \{p+1, \dots, n\}$. Possibly by multiplying the equation $a^\mathsf{T} x = \beta$ by the least common multiple of the denominators of the entries of a, we may assume that a is an integral vector. Possibly by dividing the equation $a^\mathsf{T} x = \beta$ by the greatest common divisor of the entries of a, we may assume that a has relatively prime entries. If $\beta \notin \mathbb{Z}$, then $\mathcal{S}$ is empty and we are done. Thus, we now assume $\beta \in \mathbb{Z}$. Since $a_1, \dots, a_p$ are relatively prime, by Corollary 1.9 in [2], the equation $\sum_{j=1}^p a_j x_j = \beta$ has an integral solution $\tilde{x} \in \mathbb{Z}^p$, thus $\mathcal{S}$ is nonempty. Furthermore, there exists a unimodular matrix $U \in \mathbb{Z}^{p\times p}$ such that $\tilde{a}^\mathsf{T} U = {e_1}^\mathsf{T}$, where $\tilde{a}$ is the vector of the first p coordinates of a, and e_1 denotes the first unit vector in $\mathbb{R}^p$. From the proof of Corollary 1.9 in [2], both $\tilde{x}$ and U can be computed in polynomial time. If we define the matrix $N \in \mathbb{Z}^{p\times(p-1)}$ formed by the last $p-1$ columns of U, we have

$$\left\{x \in \mathbb{Z}^p : \sum_{j=1}^p a_j x_j = \beta\right\} = \{\tilde{x} + Ny : y \in \mathbb{Z}^{p-1}\}.$$

We define the vector $\bar{x} \in \mathbb{Z}^n$ by $\bar{x}_j := \tilde{x}_j$ for $j \in \{1, \dots, p\}$ and $\bar{x}_j := 0$ for $j \in \{p+1, \dots, n\}$. We also define the matrix $M \in \mathbb{Q}^{n\times n-1}$ with block corresponding to the first p rows and $p-1$ columns being equal to N, block corresponding to the last

$n-p$ rows and $n-p$ columns being equal to the identity matrix I_{n-p}, and remaining entries zero. Since $a_i = 0$ for all $i \in \{p+1, \dots, n\}$, we conclude

$$\mathcal{S} = \{\bar{x} + My : y \in \mathbb{Z}^{p-1} \times \mathbb{R}^{n-p}\}.$$

□

Lemma 7 *Consider an instance of (MIQP) with a nonempty feasible region. There is a polynomial time algorithm that determines whether $\{x \in \mathbb{R}^n : Wx \le w\}$ is full-dimensional. If not, it rewrites the instance as an instance of (MIQP) with one fewer variable.*

Proof It is well-known [2] that there is a polynomial time algorithm that determines whether $\{x \in \mathbb{R}^n : Wx \le w\}$ is full-dimensional, and if not, finds a rational hyperplane $\{x \in \mathbb{R}^n : a^{\mathsf{T}}x = \beta\}$ that contains it. In the latter case, we let $\bar{x} \in \mathbb{Q}^n$ and $M \in \mathbb{Q}^{n\times(n-1)}$ from Lemma 6, and we define $H' := M^{\mathsf{T}}HM$, $h' := 2M^{\mathsf{T}}H^{\mathsf{T}}\bar{x} + M^{\mathsf{T}}h$, $c := \bar{x}^{\mathsf{T}}H\bar{x} + h^{\mathsf{T}}\bar{x}$, $W' := WM$, $w' := w - W\bar{x}$. By Lemma 6, our instance of (MIQP) can be rewritten as

$$\begin{aligned} \min \quad & x^{\mathsf{T}}H'x + h'^{\mathsf{T}}x + c \\ \text{s.t.} \quad & W'x \le w' \\ & x \in \Lambda, \end{aligned}$$

where

$$\Lambda := \begin{cases} \mathbb{Z}^p \times \mathbb{R}^{n-p-1} & \text{if } a_i \neq 0 \text{ for some } i \in \{p+1, \dots, n\} \\ \mathbb{Z}^{p-1} \times \mathbb{R}^{n-p} & \text{if } a_i = 0 \text{ for all } i \in \{p+1, \dots, n\}. \end{cases}$$

□

6.1 Description of the approximation algorithm

We are now in a position to present our approximation algorithm for (MIQP). We will make use of Proposition 2, Proposition 3, Proposition 5, and Lemma 7.

The input of the algorithm consists of an instance of a bounded MIQP. Theorem 4 in [10] implies that, if there is an optimal solution, there is one of size bounded by an integer ψ, which is polynomial in the size of the input MIQP. [1] Therefore, we obtain an equivalent MIQP instance by restricting each variable to the segment $[-2^{\psi}, 2^{\psi}]$. The size of the latter instance is polynomial in the size of the former. Furthermore, it is simple to check that an ϵ-approximate solution to the latter instance is also an ϵ-approximate solution to the former, for every $\epsilon \in [0, 1]$. Therefore, we can now assume that our input MIQP has a bounded feasible region.

[1] Even though Theorem 4 in [10] does not give ψ explicitly, a formula for ψ, as a function of the size of the MIQP instance, can be derived from its proof.

We initialize the set $\mathscr{I}$ of MIQP instances to be solved as a set containing only our input MIQP, and the set of possible approximate solutions as $\mathscr{A} := \emptyset$. Throughout the algorithm, each instance in $\mathscr{I}$ will be our input MIQP with a number of additional linear equality constraints. On the other hand, the set $\mathscr{A}$ will contain a number of feasible solutions to the input MIQP.

Step 1: Output, feasibility, full-dimensionality, and linear case.
Output. If $\mathscr{I} = \emptyset$, then we return the solution in $\mathscr{A}$ with the minimum objective function value if $\mathscr{A} \neq \emptyset$, and we return "infeasible" if $\mathscr{A} = \emptyset$. Otherwise $\mathscr{I} \neq \emptyset$, we choose a MIQP instance in $\mathscr{I}$ and we remove it from $\mathscr{I}$. Without loss of generality, the chosen MIQP instance is (MIQP).
Feasibility. Using Lenstra's algorithm [25], we check if the feasible region $\{x \in \mathbb{Z}^p \times \mathbb{R}^{n-p} : Wx \leq w\}$ of (MIQP) is the emptyset. If so, we go back to Step 1. Otherwise, (MIQP) is feasible and we continue.
Full-dimensionality. We apply recursively Lemma 7 until the polyhedron describing the feasible region is full-dimensional. For ease of notation, we denote the obtained instance again by (MIQP), and we now assume that $\{x \in \mathbb{R}^n : Wx \leq w\}$ is full-dimensional.
Linear case. Let k be the rank of the matrix H. If $k = 0$, (MIQP) is a MILP, and we find an optimal solution using Lenstra's algorithm. We construct the corresponding feasible solution to the input MIQP by inverting the linear transformation just performed in "Full-dimensionality", and we add it to $\mathscr{A}$. Otherwise, we have $k \geq 1$ and we continue.

Step 2: Reduction to spherical form.
By Proposition 2, we perform a change of basis that transform (MIQP) in spherical form (S-MIQP), where d satisfies $d \leq k + p$, the rank of the matrix D is k, and r_d in (9) is the ceiling of $2d^{3/2}\lceil(5d)^{d/2}\rceil^2$.

Let $B \in \mathbb{Q}^{d \times p}$ be the obtained basis matrix of the lattice Λ. By Proposition 5, we either find two aligned vectors y^+, y^- for (S-MIQP), or we find a vector $v \in \operatorname{span}(\Lambda)$ with $v^\mathsf{T} B$ integer such that $\operatorname{width}_v(\mathcal{P}) \leq r_d s_p$, where $s_p = 14p2^{p(p-1)/4}$. In the first case, continue with Step 3; In the second case, go to Step 4.

Step 3: Mesh partition and linear underestimators.
By Proposition 3 we obtain an ϵ-approximate solution $(y^\diamond, z^\diamond)$ to (S-MIQP). This requires solving, with Lenstra's algorithm, at most $\left\lceil 4r_d\sqrt{k/(3\epsilon)}\right\rceil^k$ MILPs of the same size as (S-MIQP) and with p integer variables. We construct the corresponding ϵ-approximate solution $x^\diamond$ to the (MIQP) chosen at the beginning of this iteration of the algorithm by inverting the linear transformations in Step 2 and in Step 1, and we add it to $\mathscr{A}$. Then, we go back to Step 1.

Step 4: Decomposition.
Since $\operatorname{width}_v(\mathcal{P}) \leq r_d s_p$, each point $(y, z) \in \mathcal{P}$ with $y \in \Lambda + \operatorname{span}(\Lambda)^\perp$ is contained in one of the following polytopes:

$$\mathcal{P}_t := \{(y, z) \in \mathcal{P} : v^\mathsf{T} y = t\} \qquad \text{for } t = \lceil\mu\rceil, \ldots, \lfloor\mu + r_d s_p\rfloor,$$

where $\mu := \min\{v^\mathsf{T} y : y \in \mathcal{P}\}$.

For each $t = \lceil \mu \rceil, \dots, \lfloor \mu + r_d s_p \rfloor$, we consider the instance obtained from (S-MIQP) by replacing the polytope $\mathcal{P}$ with $\mathcal{P}_t$, and we add to $\mathscr{I}$ the MIQP obtained by inverting the linear transformations in Step 2 and in Step 1. Note that the instances that we just added to $\mathscr{I}$ differ from the one chosen at the beginning of this iteration of the algorithm only by the additional constraint obtained from $v^{\mathsf{T}} y = t$ by inverting the two linear transformations. Finally, we go back to Step 1.

6.2 Analysis of the algorithm

First, we show that the algorithm described in Sect. 6.1 is correct.

Claim 3 *The algorithm in Sect. 6.1 returns an ϵ-approximate solution, if it exists.*

Proof Clearly, if the input MIQP is infeasible, the algorithm correctly detects it in Step 1, thus we now assume that it is feasible. In this case, we need to show that the algorithm returns an ϵ-approximate solution to the input MIQP. To prove this, we only need to show that the algorithm eventually adds to the set $\mathscr{A}$ an ϵ-approximate solution x^ϵ to the input MIQP. In fact, the vector returned at the end of the algorithm has objective value at most that of x^ϵ, and so it is an ϵ-approximate solution to the input MIQP as well.

Let $x^* \in \mathbb{R}^n$ be an optimal solution to the input MIQP. Let MIQP* be an instance stored at some point in $\mathscr{I}$ that contains in the feasible region the vector x^*. Among all these possible instances, we assume that MIQP*, after the “Full-dimensionality” transformation in Step 1, has a minimal number of integer variables. Note that MIQP* does not get decomposed in Step 4. Otherwise, the vector x^* would be feasible for one of the instances generated in Step 4 from MIQP*, which after the “Full-dimensionality” transformation will have fewer integer variables than MIQP*. Hence, when the algorithm selects MIQP* from $\mathscr{I}$, it performs Step 3 of the algorithm, and so by Proposition 3 it adds to $\mathscr{A}$ a vector x^ϵ that is an ϵ-approximate solution to MIQP*. Since the feasible region of MIQP* is contained in the feasible region of the input MIQP, and since the vector x^* is feasible for MIQP*, it is simple to check that x^ϵ is an ϵ-approximate solution to the input MIQP. □

We complete the proof of Theorem 1 by showing that the running time of the algorithm matches the one stated in Theorem 1.

Claim 4 *The running time of the algorithm in Sect. 6.1 is polynomial in the size of the input and in $1/\epsilon$, provided that the rank k of the matrix H and the number of integer variables p are fixed numbers.*

Proof First, we show that the algorithm performs at most $(r_{k+p} s_p + 1)^{p+1}$ iterations, which is a fixed number if both k and p are fixed. Note that the number of iterations coincides with the total number of instances that are stored in $\mathscr{I}$ throughout the execution of the algorithm. Instances are added to $\mathscr{I}$ only in Step 4, where the MIQP chosen in that iteration gets replaced in $\mathscr{I}$ with at most $r_{k'+p'} s_{p'} + 1$ new instances. Here, k' denotes the rank of the quadratic objective and p' denotes the number of integer variables of the chosen instance after the “Full-dimensionality” transformation in Step 1. In the new instances added to $\mathscr{I}$, the rank of the quadratic objective is at most k', and the number of integer variables is at most $p' - 1$. In particular, this implies that for every chosen instance we have $k' \leq k$ and $p' \leq p$. Finally, note that Step 4 may be triggered only if $p' \geq 1$. Therefore, the total number of MIQPs that are eventually stored in $\mathscr{I}$ is at most $\sum_{j=0}^{p}(r_{k+p}s_p + 1)^j \leq (r_{k+p}s_p + 1)^{p+1}$.

It is simple to check that each instance constructed by the algorithm and each number generated has size polynomial in the size of the input MIQP. Thus, to conclude the proof we only need to analyze the running time of a single iteration of the algorithm. Each MILP encountered (in Step 1 and Step 3) has at most p integer variables. Since p is fixed, they can be solved with Lenstra's algorithm [25] in time polynomial in the size of the input MIQP. Step 1 of the algorithm can then be performed in time polynomial in the size of the input MIQP. By Proposition 2 and Proposition 5, also Step 2 can be performed in time polynomial in the size of the input MIQP. In Step 3, the algorithm solves at most $\left\lceil 4r_{k+p}\sqrt{k/(3\epsilon)}\right\rceil^k$ MILPs with at most p integer variables. Since k and p are fixed, this number is polynomial in $1/\epsilon$. Therefore, Step 3 of the algorithm can be performed in time polynomial in the size of the input MIQP and in $1/\epsilon$. Step 4 only solves one linear program to find μ and stores at most $r_{k+p}s_p + 1$ MIQPs, which is a fixed number if both k and p are fixed. □

Funding: A. Del Pia is partially funded by ONR grant N00014-19-1-2322. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Office of Naval Research.

References

1. Bellare, M., Rogaway, P.: The complexity of approximating a nonlinear program. Math. Program. **69**, 429–441 (1995)
2. Conforti, M., Cornuéjols, G., Zambelli, G.: Integer Programming. Springer, Berlin (2014)
3. Cook, W., Hartman, M., Kannan, R., McDiarmid, C.: On integer points in polyhedra. Combinatorica **12**(1), 27–37 (1992)
4. Cook, W., Kannan, R., Schrijver, A.: Chvátal closures for mixed integer programming problems. Math. Program. **47**(1–3), 155–174 (1990)
5. Dax, A., Kaniel, S.: Pivoting techniques for symmetric Gaussian elimination. Numer. Math. **28**, 221–241 (1977)
6. De Loera, J., Hemmecke, R., Köppe, M., Weismantel, R.: FPTAS for optimizing polynomials over the mixed-integer points of polytopes in fixed dimension. Math. Program. Ser. A **118**, 273–290 (2008)
7. Del Pia, A.: On approximation algorithms for concave mixed-integer quadratic programming. In: Proceedings of IPCO, Lecture Notes in Computer Science, vol. 9682, pp. 1–13 (2016)
8. Del Pia, A.: On approximation algorithms for concave mixed-integer quadratic programming. Math. Program. Ser. B **172**(1–2), 3–16 (2018)
9. Del Pia, A.: Subdeterminants and concave integer quadratic programming. SIAM J. Optim. **29**(4), 3154–3173 (2019)
10. Del Pia, A., Dey, S., Molinaro, M.: Mixed-integer quadratic programming is in NP. Math. Program. Ser. A **162**(1), 225–240 (2017)
11. Del Pia, A., Weismantel, R.: Integer quadratic programming in the plane. In: Proceedings of SODA, pp. 840–846 (2014)
12. Edmonds, J.: Systems of distinct representatives and linear algebra. J. Res. Natl. Bureau Stand. B. Math. Math. Phys. **71B**(4), 241–245 (1967)
13. Eiben, E., Ganian, R., Knop, D., Ordyniak, S.: Solving integer quadratic programming via explicit and structural restrictions. Proceedings of the AAAI Conference on Artificial Intelligence (2019)

14. Galbraith, S.: Mathematics of Public Key Cryptography. Cambridge University Press, Cambridge (2012)
15. Garey, M., Johnson, D., Stockmeyer, L.: Some simplified NP-complete graph problems. Theoret. Comput. Sci. **1**(3), 237–267 (1976)
16. Golub, G., Van Loan, C.: Matrix Computations, 4th edn. Johns Hopkins University Press, Baltimore (2013)
17. Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization. Springer, Berlin (1988)
18. Hildebrand, R., Oertel, T., Weismantel, R.: Note on the complexity of the mixed-integer hull of a polyhedron. Oper. Res. Lett. **43**, 279–282 (2015)
19. Hildebrand, R., Weismantel, R., Zemmer, K.: An FPTAS for minimizing indefinite quadratic forms over integers in polyhedra. In: Proceedings of SODA, pp. 1715–1723 (2016)
20. Hochbaum, D., Shanthikumar, J.: Convex separable optimization is not much harder than linear optimization. J. Assoc. Comput. Mach. **37**(4), 843–862 (1990)
21. Khachiyan, L.: Convexity and complexity in polynomial programming. In: Proceedings of the International Congress of Mathematicians, pp. 16–24. Warsaw (1983)
22. de Klerk, E., Laurent, M., Parrilo, P.: A PTAS for the minimization of polynomials of fixed degree over the simplex. Theoret. Comput. Sci. **361**, 210–225 (2006)
23. Kozlov, M., Tarasov, S., Khachiyan, L.: Polynomial solvability of convex quadratic programming. Doklady Akademii Nauk SSSR **248**, 1049–1051 (1979). Translated in: Soviet Mathematics Doklady 20 (1979) 1108-1111
24. Lee, J., Onn, S., Romanchuk, L., Weismantel, R.: The quadratic graver cone, quadratic integer minimization, and extensions. Math. Program. Ser. B **136**, 301–323 (2012)
25. Lenstra, H.J.: Integer programming with a fixed number of variables. Math. Oper. Res. **8**(4), 538–548 (1983)
26. Murty, K., Kabadi, S.: Some NP-complete problems in quadratic and linear programming. Math. Program. **39**, 117–129 (1987)
27. Nemirovsky, A., Yudin, D.: Problem Complexity and Method Efficiency in Optimization. Wiley, Chichester (1983). Translated by E.R. Dawson from Slozhnost' Zadach i Effektivnost' Metodov Optimizatsii (1979)
28. Pardalos, P., Vavasis, S.: Quadratic programming with one negative Eigenvalue is NP-hard. J. Glob. Optim. **1**(1), 15–22 (1991)
29. Vavasis, S.: Quadratic programming is in NP. Inf. Process. Lett. **36**, 73–77 (1990)
30. Vavasis, S.: Approximation algorithms for indefinite quadratic programming. Math. Program. **57**, 279–311 (1992)
31. Vavasis, S.: On approximation algorithms for concave quadratic programming. In: Floudas, C., Pardalos, P. (eds.) Recent Advances in Global Optimization, pp. 3–18. Princeton University Press, Princeton, NJ (1992)
32. Vavasis, S.: Polynomial time weak approximation algorithms for quadratic programming. In: P. Pardalos (ed.) Complexity in Numerical Optimization. World Scientific (1993)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.